# **Control Unit**

One of the basic modules for a CPU is the control unit. Our task was to design the control unit with the following functions: **sw, lw, beq, bne, j, add, sub, and, or, slt, addi.** The implementation provided by lecturer had almost all the functionality implemented except: **j, bne, addi.** From the MIPS32 instruction sheet we found out the corresponding OP codes for the missing instructions.

Instruction	OP Code (Hex)
J	0x02
Bne	0x05
addi	0x08

J and **addi** was very easy to integrate in the current design. We were able to use the existing hardware design, however for the **bne** instruction we had to add some more components to the design. Basically we added another branch data path, called "BranchNE" to the control unit.

## Simulation

After we had described the behaviour of the control unit we simulated it with the OP codes for the previously mentioned instructions.

		2.436 ns											69.428 ns
Name	Value		10 n:	s  20 ns		30 ns	s 	40 n:	s 	50 n	s 	60 ns	
🕨 📑 opcode[5:0]	08	00		23 X	2b X		04		02	Х —	05	$\square$	08
🕨 📑 aluop[1:0]	0	2	$\sim$	0	X		1	$\square$	0	Х—	1	$\square$	0
🗓 regdst	0												
🗓 alusrc	1												
🐚 memtoreg	0												
🗓 regwrite	1												
🗓 memread	0												
🗓 memwrite	0												
∿ jump	0												
U branchne	0												
1 branch	0												

### Result

- 0-10ns is R-type instruction
- 10-20ns is load word instruction
- 20-30ns is store word instruction
- 30-40ns is branch if equals instruction
- 40-50ns is jump instruction
- 50-60ns is branch not equals instruction
- 60-70ns is addi instruction

## Conclusion

The simulation of the control unit worked just like expected. All the corresponding bits regarding the OP code were set correctly as you can see in the simulation waveform above.

# Arithmetic logic unit (ALU)

After we had a working control unit next task was to implement ALU for all the logical operations regarding registers, like adding, subtracting, anding, oring etc. Again the lecturer provided us with a sample ALU VHDL code. This time the provided code had all the functionality necessary for ALU, so we didn't need to alter it at all. However, we did split the ALU in 3 parts – Sign extender, ALU control unit and ALU, so our final implementation would look like the provided schematic for MIPS32 processor.

## ALU control unit

We implemented ALU control unit by following "ControlUnit.docx" document provided by our lecturer. Basically the task of the ALU control unit is to specify what type of arithmetic operation the main ALU has to carry out.

Logical /Arithmetic Operation	ALUCtr[2:0]
and	000
or	001
add	010
sub	110
slt	111

Depending on the ALUOp bus provided by Control Unit and the instruction bus we had to implement the following truth table in ALU control unit

opcode	ALUOp	Operation	instr	ALU function	ALUCtr[2:0]
lw	00	Load word	хххххх	Add	010
SW	00	Store word	XXXXXX	Add	010
beq	01	Branch if equal	хххххх	subtract	110
R_type	10	Add	100000	Add	010
		subtract	100010	Subtract	110
		AND	100100	AND	000
		OR	100101	OR	001
		SLT	101010	Set on less than	111

## Simulation

		1.218 ns											69.428 ns
Name	Value		10 1	ns	20	ns	30	ns  4	0 ns	50	ns	60	ns
aluop[1:0]	10	00		01	Х				10				
instr[5:0]	101010		000000	)	ΧС	100000	$\sim$	100010 🗙	100100	х	100101	Х	101010
aluctr[2:0]	111	010		110	ΧЦ	010	$\sim$	110 X	000	х	001	Х	111
	1												

### Results

- 0-10ns load/store word instruction
- 10-20ns branch instruction
- 20-30ns add instruction
- 30-40ns subtract instruction
- 40-50ns AND instruction
- 50-60ns OR instruction
- 60-70ns SLT instruction

### Conclusion

We can confirm that the simulation worked just like expected. The ALU Control unit generated correct output for every input just like described in the truth table.

### Main ALU

The actual arithmetic operations are carried out in the main ALU. For our convenience in this module we integrated one of the MUX provided in the schematic. The one which selects between register files "Read Data 2" output and sign extender output. Once again, the code was more or less provided by the lecturer.

### Simulation

We simulated whether the integrated MUX and all the arithmetic functions are working in the ALU.

		85.262 ns										180.268 ns
Name	Value	100 ns			120 ns			140 ns		<u> </u>	160 ns	
🕨 📲 data1[31:0]	000000aa	0000001		0000	0003		X 000000a	0000	0003	00000010	Х	000000aa
🕨 📲 data2[31:0]	000000aa	0000002		0000007	0000	000a	0000003	0000	0010	0000003	X	000000aa
sign_extended[31:0]	0000003						00000003					
Un alusrc	0											
aluctr[2:0]	6	2		0	( 1		6	×		7	X	6
Uo zero	1											
alu_result[31:0]	00000000	0000003 X 0000	0004	0000003	0000	000b	0000007	0000	0001	00000000	X	0000000

### Result

- 85 100ns select data2 as input and perform addition
- 100 110ns select sign extend as input and perform addition
- 110 120ns perform AND operation
- 120 130ns perform OR operation
- 130 140ns perform subtraction
- 140 150ns perform STL operation

- 150-160ns perform STL operation with reversed input
- 160-170ns perform subtraction with identical inputs

### Conclusion

Everything worked just like expected. The ALU carried out all the arithmetic operations correctly, and the integrated MUX was performing correctly as well.

## Sign Extender

One last component we have to design is sign extender before we can combine the actual CPU. Out lecturer provided with an example sign extender for 8 bit data bus. Only thing we had to change is the BUS width to 32 bits. For our sign extender we used the IEEE numeric\_std library.

### Simulation

Sign extension HEX result



Sign extension Decimal result

		0.000 ns							40.051 ns
Name	Value	0 ns	5 ns	10 ns	15 ns	20 ns	25 ns	30 ns	35 ns
input[15:0]	-6			-32	768	1	1	-	6
output[31:0]	-6			-32	768	17	1	-	6
a set a s	1								

### Result

- 0 10ns extended number 0x0001 to 0x00000001
- 10 20ns extended number 0x8000 to 0xffff8000
- 20 30ns extended number 0x00ab to 0x000000ab
- 30 40ns extended number 0xfffa to 0xffffffa

### Conclusion

As you can see in the two waveforms shown above, the sign extensions between 2 byte number and 8 byte number is working correctly.

# **Single Cycle Processor implementation**

At this stage we have designed all the necessary components for single cycle processor. Our next and final task is to merge everything together and simulate the CPU. We used the following schematic for our implementation:



However, as previously mentioned we slightly altered the branching to implement the BNE instruction:



## Simulation

After we assembled the CPU with structural VHDL, we then needed a machine code to test whether our implementation and all the components together are working correctly. We designed a test code on MARS, and then just copied the machine code over to our instruction memory.

Only thing we have to alter is the jump address when executing J instruction. MARS starts the code from address 0x00400000. We start the code from the 0<sup>th</sup> address. BNE or BEQ works fine because they are not jumping to a specific address, but it alters the PC value by the difference between desired address and current address.

### 1<sup>st</sup> test code

We wrote a small test code which would test all the instructions our implementation can handle.

#### Code in MARS

```
1
2 number:
                   .word 0
3
    .text
4
5 main:
           addi $t1, $0, 100
6
7
           sw $t1, number
           lw $t2, number
8
9
          beq $t1, $t2, load2
10
           addi $t3, $0, 1
11
           j skipl
12
13
14 load2: addi $t3, $0, 2
15
   skip1: bne $t1, $t3, subtract
16
17
           add $t0, $t1, $t3
18
           j skip2
19
20 subtract:
21
           sub $t0, $t1, $t3
22
23 skip2: bne $tl, $t2, main
24
           add $t4, $t1, $t2
25
           and $t5, $t1, $t0
26
27
           or $t6, $t1, $t0
28
29
           j setless
           addi $t7, $t7, 10000
30
31
32 setless:
           slt $t7, $t5, $t1
33
34
35
36
```

Name         Number         Value           \$zero         0         0x0000000           \$at         1         0x10010000           \$v0         2         0x0000000           \$v1         3         0x0000000           \$a0         4         0x0000000           \$a1         5         0x0000000           \$a1         5         0x0000000           \$a2         6         0x0000000           \$a3         7         0x0000000           \$a1         5         0x0000000           \$a2         6         0x0000000           \$a3         7         0x0000000           \$a1         5         0x0000000           \$a2         10         0x0000000           \$t1         9         0x0000000           \$t2         10         0x0000000           \$t4         12         0x0000000           \$s0         16         0x0000000           \$s1         17         0x0000000           \$s2         18         0x0000000           \$s4         20         0x0000000           \$s5         21         0x0000000           \$s7         23	Registers	Coproc 1	Coproc 0		
\$zero         0         0x0000000           \$xt         1         0x10010000           \$v0         2         0x0000000           \$v1         3         0x0000000           \$a0         4         0x0000000           \$a1         5         0x0000000           \$a2         6         0x0000000           \$a3         7         0x0000000           \$a3         7         0x0000000           \$a4         0x0000000         8           \$b1         9         0x0000000           \$a3         7         0x0000000           \$b1         9         0x0000000           \$b2         10         0x0000000           \$b1         9         0x0000000           \$b1         0x0000000         8           \$b2         13         0x0000000           \$b2         13         0x0000000           \$b2         13         0x0000000           \$b2         13         0x0000000           \$b2         18         0x0000000           \$b2         18         0x0000000           \$b2         2         0x0000000           \$b2         2	N	ame	N	lumber	Value
\$at         1         0x10010000           \$v0         2         0x00000000           \$v1         3         0x0000000           \$a0         4         0x0000000           \$a1         5         0x0000000           \$a2         6         0x0000000           \$a3         7         0x0000000           \$a3         7         0x0000000           \$t1         9         0x00000062           \$t1         9         0x0000000           \$t2         10         0x0000000           \$t2         10         0x00000000           \$t2         00         0x0000000           \$t2         10         0x0000000           \$t2         0x00000000         \$t4           \$t3         11         0x0000000           \$s0         15         0x0000000           \$s1         17         0x0000000           \$s2         18         0x0000000           \$s4         20         0x0000000           \$s5         21         0x0000000           \$s6         22         0x0000000           \$s6         22         0x00000000           \$k0         25 </td <td>\$zero</td> <td></td> <td></td> <td>0</td> <td>0x0000000</td>	\$zero			0	0x0000000
\$v0         2         0x0000000           \$v1         3         0x0000000           \$a0         4         0x0000000           \$a1         5         0x0000000           \$a2         6         0x0000000           \$a3         7         0x0000000           \$a3         7         0x0000000           \$a3         7         0x0000000           \$a1         9         0x0000000           \$c1         9         0x0000000           \$c1         9         0x0000000           \$c1         9         0x0000000           \$c1         9         0x0000000           \$c2         10         0x00000000           \$c4         12         0x0000000           \$c5         13         0x0000000           \$c6         14         0x0000000           \$c7         15         0x0000000           \$c8         21         0x0000000           \$c8         22         0x0000000           \$c7         23         0x0000000           \$c8         22         0x0000000           \$c7         23         0x0000000           \$c8         0x00000000 <td>\$at</td> <td></td> <td></td> <td>1</td> <td>0x10010000</td>	\$at			1	0x10010000
\$v1         3         0x0000000           \$a0         4         0x0000000           \$a1         5         0x0000000           \$a2         6         0x0000000           \$a3         7         0x0000000           \$a1         9         0x0000000           \$a2         6         0x0000000           \$a3         7         0x0000000           \$a1         9         0x0000000           \$t0         8         0x0000000           \$t1         9         0x0000000           \$t2         10         0x0000000           \$t4         12         0x0000000           \$t5         13         0x0000000           \$t6         14         0x0000000           \$s1         17         0x0000000           \$s2         18         0x0000000           \$s4         20         0x0000000           \$s5         21         0x0000000           \$s6         22         0x0000000           \$s7         23         0x0000000           \$k1         27         0x0000000           \$k2         0x00000000         0x10           \$k2         0x0000000	\$v0			2	0x0000000
\$a0         4         0x0000000           \$a1         5         0x0000000           \$a2         6         0x0000000           \$a3         7         0x0000000           \$t0         8         0x0000000           \$t1         9         0x0000000           \$t2         10         0x0000000           \$t4         12         0x0000000           \$t5         13         0x0000000           \$t6         14         0x0000000           \$t7         15         0x0000000           \$s1         17         0x0000000           \$s2         18         0x0000000           \$s4         20         0x0000000           \$s5         21         0x0000000           \$s7         23         0x0000000           \$s7         23         0x0000000           \$k0         26         0x0000000           \$k1         27         0x0000000           \$k2         0x0000000         0x10000000           \$k1         27         0x0000000           \$k2         0x0000000         0x10000000           \$k4         20         0x00000000           \$k5	\$v1			3	0x0000000
\$a1         5         0x0000000           \$a2         6         0x0000000           \$a3         7         0x0000000           \$t0         8         0x00000064           \$t1         9         0x00000064           \$t2         10         0x00000064           \$t1         9         0x00000064           \$t2         10         0x0000000           \$t4         11         0x00000066           \$t5         13         0x00000066           \$t5         13         0x0000000           \$t6         14         0x0000000           \$t7         15         0x0000000           \$s1         17         0x0000000           \$s2         18         0x0000000           \$s3         19         0x0000000           \$s4         20         0x0000000           \$s5         21         0x0000000           \$s6         22         0x0000000           \$s7         23         0x0000000           \$s8         24         0x0000000           \$s9         25         0x0000000           \$s1         27         0x00000000           \$s2	\$a0			4	0x0000000
\$a2         6         0x0000000           \$a3         7         0x0000000           \$t0         8         0x0000062           \$t1         9         0x00000062           \$t2         10         0x0000000           \$t2         10         0x0000000           \$t4         11         0x0000000           \$t5         13         0x0000000           \$t6         14         0x0000000           \$t6         14         0x0000000           \$t7         15         0x0000000           \$s0         16         0x0000000           \$s1         17         0x0000000           \$s2         18         0x0000000           \$s5         21         0x0000000           \$s5         21         0x0000000           \$s6         22         0x0000000           \$s7         23         0x0000000           \$s4         20         0x0000000           \$s7         23         0x0000000           \$s7         23         0x0000000           \$s9         25         0x0000000           \$s1         27         0x00000000           \$s1         27<	\$a1			5	0x0000000
\$43         7         0x0000000           \$t0         8         0x00000062           \$t1         9         0x00000064           \$t2         10         0x00000002           \$t3         11         0x00000002           \$t4         12         0x00000006           \$t5         3         0x00000000           \$t6         14         0x0000000           \$t6         14         0x0000000           \$t7         15         0x0000000           \$s1         17         0x0000000           \$s2         18         0x0000000           \$s3         19         0x0000000           \$s5         21         0x0000000           \$s6         22         0x0000000           \$s7         23         0x0000000           \$s6         24         0x0000000           \$s7         25         0x0000000           \$k1         27         0x0000000           \$k2         30         0x0000000           \$k1         27         0x0000000           \$k2         0x0000000         0x10000000           \$k1         27         0x00000000           \$k2	\$a2			6	0x0000000
\$t0         8         0x0000062           \$t1         9         0x0000064           \$t2         10         0x0000064           \$t3         11         0x0000002           \$t4         12         0x00000066           \$t5         13         0x00000066           \$t6         14         0x0000000           \$t7         15         0x0000000           \$s0         16         0x0000000           \$s1         17         0x0000000           \$s2         18         0x0000000           \$s3         19         0x0000000           \$s4         20         0x0000000           \$s5         21         0x0000000           \$s6         22         0x0000000           \$s7         23         0x0000000           \$s6         24         0x0000000           \$s7         25         0x0000000           \$s9         25         0x0000000           \$s9         29         0x7fffeffc           \$spp         30         0x0000000           \$ra         31         0x0000000           \$ra         31         0x0000000      \$ra         0x00000000<	\$a3			7	0x0000000
\$t1         9         0x0000064           \$t2         10         0x0000002           \$t3         11         0x0000002           \$t4         12         0x0000006           \$t5         13         0x0000066           \$t6         14         0x0000006           \$t7         15         0x0000000           \$s0         16         0x0000000           \$s1         17         0x0000000           \$s2         18         0x0000000           \$s3         19         0x0000000           \$s4         20         0x0000000           \$s5         21         0x0000000           \$s7         23         0x0000000           \$t8         24         0x0000000           \$t9         25         0x0000000           \$t1         27         0x0000000           \$spp         29         0x1fffeffef           \$spp         29         0x1fffeffef           \$spp         30         0x0000000           \$ra         31         0x0000000           \$ra         31         0x0000000           \$ra         31         0x0000000 <tr tbody=""></tr>	\$t0			8	0x0000062
\$t2         10         0x0000064           \$t3         11         0x0000002           \$t4         12         0x0000066           \$t5         13         0x0000066           \$t6         14         0x0000006           \$t6         14         0x0000000           \$s0         16         0x0000000           \$s1         17         0x0000000           \$s2         18         0x0000000           \$s3         19         0x0000000           \$s4         20         0x0000000           \$s5         21         0x0000000           \$s6         22         0x0000000           \$s7         23         0x0000000           \$s7         23         0x0000000           \$t8         24         0x0000000           \$t9         25         0x0000000           \$k0         26         0x0000000           \$s1         30         0x0000000           \$k1         27         0x0000000           \$k2         31         0x0000000           \$sp         29         0x7ffeffc           \$fp         30         0x00000000      \$ra         31	\$t1			9	0x0000064
\$t3         11         0x0000002           \$t4         12         0x0000006           \$t5         13         0x0000066           \$t6         14         0x0000006           \$t7         15         0x0000000           \$s0         16         0x0000000           \$s1         17         0x0000000           \$s2         18         0x0000000           \$s3         19         0x0000000           \$s4         20         0x0000000           \$s5         21         0x0000000           \$s6         22         0x0000000           \$s5         21         0x0000000           \$s6         22         0x0000000           \$s7         23         0x0000000           \$s7         23         0x0000000           \$s7         25         0x0000000           \$s8         24         0x0000000           \$s9         25         0x0000000           \$s1         30         0x0000000           \$s1         30         0x0000000           \$s2         30         0x0000000           \$s1         30         0x00000000           \$s2	\$t2			10	0x0000064
\$t4         12         0x000000c8           \$t5         13         0x0000006           \$t6         14         0x0000006           \$t7         15         0x0000000           \$s0         16         0x0000000           \$s1         17         0x0000000           \$s2         18         0x0000000           \$s3         19         0x0000000           \$s5         21         0x0000000           \$s6         22         0x0000000           \$s7         23         0x0000000           \$s7         23         0x0000000           \$t8         24         0x0000000           \$t9         25         0x0000000           \$k0         26         0x0000000           \$k1         27         0x0000000           \$k2         30         0x0000000           \$k1         27         0x0000000           \$k2         0x0000000         \$k1           \$k2         0x0000000         \$k2           \$k1         0x0000000         \$k2           \$k2         0x0000000         \$k2           \$k2         0x0000000         \$k2           \$k3	\$t3			11	0x0000002
\$t5         13         0x0000060           \$t6         14         0x0000066           \$t7         15         0x0000000           \$s0         16         0x0000000           \$s1         17         0x0000000           \$s2         18         0x0000000           \$s3         19         0x0000000           \$s4         20         0x0000000           \$s5         21         0x0000000           \$s6         22         0x0000000           \$s7         23         0x0000000           \$s7         23         0x0000000           \$s8         24         0x0000000           \$s7         23         0x0000000           \$s1         27         0x0000000           \$s2         0x0000000         \$k0         26         0x0000000           \$k0         26         0x0000000         \$k1         27         0x0000000           \$k1         27         0x0000000         \$k2         0x0000000         \$k1         0x0000000           \$s1         30         0x0000000         \$k1         0x0000000         \$k2         0x0000000           \$s2         30         0x00000000	\$t4			12	0x00000c8
\$t6         14         0x0000066           \$t7         15         0x0000000           \$s0         16         0x0000000           \$s1         17         0x0000000           \$s2         18         0x0000000           \$s3         19         0x0000000           \$s4         20         0x0000000           \$s5         21         0x0000000           \$s6         22         0x0000000           \$s7         23         0x0000000           \$s7         23         0x0000000           \$s8         24         0x0000000           \$t8         24         0x0000000           \$k0         26         0x0000000           \$k1         27         0x0000000           \$sp         29         0x7ffeffc           \$fp         30         0x0000000           \$ra         31         0x0000000           \$ra         31         0x0000000           \$ra         31         0x0000000           \$ra         31         0x0000000           \$ra         0x00000000         0x00000000	\$t5			13	0x0000060
\$t7         15         0x0000001           \$s0         16         0x0000000           \$s1         17         0x0000000           \$s2         18         0x0000000           \$s3         19         0x0000000           \$s4         20         0x0000000           \$s6         21         0x0000000           \$s7         23         0x0000000           \$t8         24         0x0000000           \$s7         23         0x0000000           \$t8         24         0x0000000           \$t9         25         0x0000000           \$k0         26         0x0000000           \$k1         27         0x0000000           \$sp         29         0x7ffeffc           \$fp         30         0x0000000           \$ra         31         0x0000000           \$ra         31         0x0000000           \$ra         0x0000000         0x00000000	\$t6			14	0x0000066
\$\$0         16         0x0000000           \$\$1         17         0x0000000           \$\$2         18         0x0000000           \$\$3         19         0x0000000           \$\$4         20         0x0000000           \$\$5         21         0x0000000           \$\$6         22         0x0000000           \$\$7         23         0x0000000           \$\$1         24         0x0000000           \$\$1         25         0x0000000           \$\$1         27         0x0000000           \$\$2         0x0000000         \$\$2           \$\$1         27         0x0000000           \$\$2         0x0000000         \$\$2           \$\$2	\$t7			15	0x0000001
\$\$1         17         0x0000000           \$\$2         18         0x0000000           \$\$3         19         0x0000000           \$\$4         20         0x0000000           \$\$5         21         0x0000000           \$\$6         22         0x0000000           \$\$7         23         0x0000000           \$\$18         24         0x0000000           \$\$19         25         0x0000000           \$\$19         25         0x0000000           \$\$10         26         0x0000000           \$\$11         27         0x0000000           \$\$29         0x7fffeffc         \$\$29           \$\$29         0x7fffeffc         \$\$29           \$\$29         30         0x0000000           \$\$29         31         0x0000000           \$\$13         0x0000000         \$\$29           \$\$29         30         0x0000000           \$\$13         0x00000000         \$\$10           \$\$14         0x00000000         \$\$10           \$\$20         0x00000000         \$\$10           \$\$20         0x00000000         \$\$10           \$\$20         0x00000000         \$\$10	\$s0			16	0x00000x0
\$\$2         18         0x0000000           \$\$3         19         0x0000000           \$\$4         20         0x0000000           \$\$5         21         0x0000000           \$\$6         22         0x0000000           \$\$7         23         0x0000000           \$\$7         23         0x0000000           \$\$7         23         0x0000000           \$\$18         24         0x0000000           \$\$19         25         0x0000000           \$\$10         27         0x0000000           \$\$11         27         0x0000000           \$\$29         0x7fffeffc         \$\$29           \$\$29         0x7fffeffc         \$\$1008000           \$\$29         30         0x0000000           \$\$29         31         0x0000000           \$\$13         0x0000000         \$\$14           \$\$10         31         0x0000000           \$\$11         0x00000000         \$\$14           \$\$10         0x00000000         \$\$15	\$s1			17	0x00000x0
\$s3         19         0x0000000           \$s4         20         0x0000000           \$s5         21         0x0000000           \$s6         22         0x0000000           \$s7         23         0x0000000           \$t8         24         0x0000000           \$t9         25         0x0000000           \$k0         26         0x0000000           \$sp         27         0x0000000           \$sp         29         0x7fffeffc           \$fp         30         0x0000000           \$ra         31         0x0000000           \$ra         31         0x0000000           \$ra         0x0000000         0x0000000           \$ra         0x0000000         0x0000000	\$s2			18	0x00000x0
\$s4         20         0x0000000           \$s5         21         0x0000000           \$s6         22         0x0000000           \$s7         23         0x0000000           \$t8         24         0x0000000           \$t9         25         0x0000000           \$k0         26         0x0000000           \$sp         27         0x0000000           \$sp         29         0x7fffeffc           \$fp         30         0x0000000           \$ra         31         0x0000000           hi         0x0000000         0x0000000	\$s3			19	0x00000x0
\$s5         21         0x0000000           \$s6         22         0x0000000           \$s7         23         0x0000000           \$t8         24         0x0000000           \$t9         25         0x0000000           \$k0         26         0x0000000           \$k1         27         0x0000000           \$sp         29         0x7fffeffc           \$fp         30         0x0000000           \$ra         31         0x0000000           hi         0x0000000         0x0000000	\$s4			20	0x00000x0
\$\$6         22         0x0000000           \$\$7         23         0x0000000           \$t8         24         0x0000000           \$t9         25         0x0000000           \$k0         26         0x0000000           \$k1         27         0x0000000           \$gp         28         0x1008000           \$sp         29         0x7fffeffc           \$fp         30         0x0000000           \$ra         31         0x0000000           hi         0x0000000         0x0000000	\$s5			21	0x00000x0
\$\$7         23         0x0000000           \$t8         24         0x0000000           \$t9         25         0x0000000           \$k0         26         0x0000000           \$k1         27         0x0000000           \$gp         28         0x10008000           \$sp         29         0x7ffefcc           \$fp         30         0x0000000           \$ra         31         0x0000000           hi         0x00000000         0x00000000	\$s6			22	0x00000x0
\$t8         24         0x0000000           \$t9         25         0x0000000           \$k0         26         0x0000000           \$k1         27         0x0000000           \$gp         28         0x10008000           \$sp         29         0x7ffeffc           \$fp         30         0x0000000           \$ra         31         0x0000000           pc         0x00000000         0x00000000           hi         0x00000000         0x00000000	\$s7			23	0x00000x0
\$t9         25         0x0000000           \$k0         26         0x0000000           \$k1         27         0x0000000           \$gp         28         0x10008000           \$sp         29         0x7fffeffc           \$fp         30         0x0000000           \$ra         31         0x0000000           pc         0x0000000         0x0000000           hi         0x00000000         0x00000000	\$t8			24	0x00000x0
\$k0         26         0x0000000           \$k1         27         0x0000000           \$gp         28         0x10008000           \$sp         29         0x7fffeffc           \$fp         30         0x0000000           \$ra         31         0x0000000           pc         0x0000000         0x0000000           hi         0x00000000         0x00000000           lo         0x00000000         0x00000000	\$t9			25	0x00000x0
\$k1         27         0x0000000           \$gp         28         0x10008000           \$sp         29         0x7ffeffc           \$fp         30         0x0000000           \$ra         31         0x0000000           \$pc         0x00000000         0x00000000           \$hi         0x00000000         0x00000000           \$loo         0x00000000         0x00000000	\$k0			26	0x00000x0
\$gp         28         0x10008000           \$sp         29         0x7ffeffc           \$fp         30         0x0000000           \$ra         31         0x0000000           pc         0x0000000         0x0000000           hi         0x0000000         0x0000000           lo         0x0000000         0x0000000	\$k1			27	0x00000x0
\$sp         29         0x7ffeffc           \$fp         30         0x0000000           \$ra         31         0x0000000           pc         0x0040050         0x0000000           hi         0x0000000         0x0000000           lo         0x0000000         0x0000000	\$gp			28	0x10008000
\$fp         30         0x0000000           \$ra         31         0x0000000           pc         0x0040050         0x0000000           hi         0x0000000         0x0000000           lo         0x0000000         0x0000000	\$sp			29	0x7fffeffc
\$ra         31         0x0000000           pc         0x0040050         0x00400050           hi         0x00000000         0x00000000           lo         0x00000000         0x00000000	\$fp			30	0x00000x0
pc 0x00400050 hi 0x0000000 lo 0x0000000	\$ra			31	0x00000x0
hi 0x0000000 lo 0x0000000	pc				0x00400050
lo 0x0000000	hi				0x0000000
	10				0x00000x0

### Register values in MARS after execution

### Machine code in Xilinx

X"20090064"	X"ac290028"	X"8c2a0028"	X"112a0002"	X"200b0001"
X"08100009"	X"200b0002"	X"152b0002"	X"012b4020"	X"0810000d"
X"012b4022"	X"152afff2"	X"012a6020"	X"01286824"	X"01287025"
X"08000011"	X"21ef2710"	X"01a9782a"		

#### Register values in Xilinx after execution

$\triangleright$	-6	at_reg[31:0]	0000000	Array
⊳	-6	a0_reg[31:0]	0000000	Array
$\triangleright$	-6	a1_reg[31:0]	0000000	Array
⊳	-6	a2_reg[31:0]	0000000	Array
$\triangleright$	-6	a3_reg[31:0]	0000000	Array
⊳	-6	E[31:0]	0000001	Array
$\triangleright$	-6	fp_reg[31:0]	0000000	Array
⊳	-0	gp_reg[31:0]	0000000	Array
$\triangleright$	-6	k0_reg[31:0]	0000000	Array
$\triangleright$	-0	k1_reg[31:0]	0000000	Array
$\triangleright$	-6	ra_reg[31:0]	0000000	Array
⊳	-0	sp_reg[31:0]	0000000	Array
$\triangleright$	-6	s0_reg[31:0]	0000000	Array
⊳	-6	s1_reg[31:0]	0000000	Array
$\triangleright$	-6	s2_reg[31:0]	0000000	Array
⊳	-6	s3_reg[31:0]	0000000	Array
$\triangleright$	-6	s4_reg[31:0]	0000000	Array
⊳	-6	s5_reg[31:0]	00000000	Array
$\triangleright$	-6	s6_reg[31:0]	0000000	Array
⊳	-0	s7_reg[31:0]	0000000	Array
$\triangleright$	-6	t0_reg[31:0]	00000062	Array
⊳		t1_reg[31:0]	00000064	Array
$\triangleright$	-0	t2_reg[31:0]	00000064	Array
⊳	-0	t3_reg[31:0]	00000002	Array
$\triangleright$	-0	t4_reg[31:0]	00000c8	Array
⊳	-0	t5_reg[31:0]	00000060	Array
$\triangleright$	-0	t6_reg[31:0]	00000066	Array
⊳	-0	t7_reg[31:0]	00000001	Array
$\triangleright$	-0	t8_reg[31:0]	0000000	Array
⊳	-0	t9_reg[31:0]	0000000	Array
$\triangleright$	-0	v0_reg[31:0]	0000000	Array
⊳	-0	v1_reg[31:0]	0000000	Array
	-0	zero_reg[31	0000000	Array

### Register value comparison between MARS and Xilinx

Register	MARS	Xilinx
\$t0	0x62	0x62
\$t1	0x64	0x64
\$t2	0x64	0x64
\$t3	0x02	0x02
\$t4	0xc8	0xc8
\$t5	0x60	0x60
\$t6	0x66	0x66
\$t7	0x01	0x01

From the first test code we can confirm that all the necessary instructions have been implemented correctly in our CPU design. We weren't confident that jumping and branching instruction will work. Despite our pessimism those two instructions were executed correctly.

### $2^{nd}$ test code

After we confirmed that all of the instructions are executed correctly, we thought of making another test code for calculating Fibannoci sequence and store it into the memory.

#### **Code in MARS**

```
1
     . data
    fibs: .word 0: 10
                                        #
 2
 3
     .text
 4
 5 main:
 6
7
                addi $t1, $0, 1
         addi $t2, $0, 1
 8
9
                addi $t3, $0, 0
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
                sw $tl, ($t3)
                addi $t3, $t3, 4
sw $t2, ($t3)
addi $t3, $t3, 4
                addi $t0, $0, 0
                add $t4, $t1, $t2
                add $t1, $0, $t2
add $t2, $0, $t4
                sw $t4, ($t3)
                addi $t3, $t3, 4
                addi $t0, $t0, 1
                bne $t0, 15, loop
31
32
```

#### Memory contents in MARS after execution

Data Segment	Data Segment 🗖										
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)			
0x10010000	1	1	2	3	5	8	13	21 🔺			
0x10010020	34	55	89	144	233	377	610	987			
0x10010040	1597	0	0	0	0	0	0	0			
0x10010060	0	0	0	0	0	0	0	0			
0x10010080	0	0	0	0	0	0	0	0			
0x100100a0	0	0	0	0	0	0	0	0			
0x100100c0	0	0	0	0	0	0	0	0 🚽			
0x100100e0	0	0	0	0	0	0	0	0			
0x10010100	0	0	0	0	0	0	0	0			
0x10010120	0	0	0	0	0	0	0	0			
0x10010140	0	0	0	0	0	0	0	0			
0x10010160	0	0	0	0	0	0	0	0			
0x10010180	0	0	0	0	0	0	0	0			
0x100101a0	0	0	0	0	0	0	0	0			
	0	0	0	0	0	0	0				
🚸 🐟 0x10010000 (.data) 🔻 🕑 Hexadecimal Addresses 🗋 Hexadecimal Values 🗔 ASCII											

Registers Coproc 1	Coproc 0				
Name	Number	Value			
\$zero	C	0			
\$at	1	. 15			
\$v0	2				
\$v1	3	0			
\$a0	4	0			
\$al	5	0			
\$a2	6	i 0			
\$a3	7	0			
\$t0	8	15			
\$t1	9	987			
\$t2	10	1597			
\$t3	11	. 268501060			
\$t4	12	1597			
\$t5	13	0			
\$t6	14	0			
\$t7	15	i 0			
\$s0	16				
\$s1	17	0			
\$s2	18	0			
\$ <b>3</b> 3	19	0			
\$34	20	0			
\$s5	21	. 0			
\$36	22	2 0			
\$s7	23	0			
\$t8	24	0			
\$t9	25	0			
\$k0	26	0			
\$k1	27	0			
\$gp	28	268468224			
şap	29	2147479548			
şfp	30	0			
şra	31	0			
pc		4194372			
hi		0			
10		0			

### Register values in MARS after execution

#### Machine code in Xilinx

X"20090001"	X"200a0001"	X"200b0000"	X"ad690000"	X"216b0004"
X"ad6a0000"	X"216b0004"	X"20080000"	X"012a6020"	X"000a4820"
X"000c5020"	X"ad6c0000"	X"216b0004"	X"21080001"	X"2001000f"
X"1428fff8"				

## Register values in Xilinx after execution

Object Name	Value	Data Type
RS_OUT[31:0]	0	Array
RT_OUT[31:0]	0	Array
at_reg[31:0]	15	Array
> 📲 a0_reg[31:0]	0	Array
All a1_reg[31:0]	0	Array
⊳ 🔏 a2_reg[31:0]	0	Array
a3_reg[31:0]	0	Array
⊳ 📲 E[31:0]	1	Array
⊳ 考 fp_reg[31:0]	0	Array
⊳ 考 gp_reg[31:0]	0	Array
⊳ 🔏 k0_reg[31:0]	0	Array
⊳ 🔏 k1_reg[31:0]	0	Array
⊳ 💑 ra_reg[31:0]	0	Array
⊳ 💑 sp_reg[31:0]	0	Array
b a s0_reg[31:0]	0	Array
s1_reg[31:0]	0	Array
> 💑 s2_reg[31:0]	0	Array
S3_reg[31:0]	0	Array
⊳ 💑 s4_reg[31:0]	0	Array
⊳ 💑 s5_reg[31:0]	0	Array
⊳ 📲 s6_reg[31:0]	0	Array
s7_reg[31:0]	0	Array
t0_reg[31:0]	15	Array
	987	Array
t2_reg[31:0]	1597	Array
b 😼 t3_reg[31:0]	68	Array
t4_reg[31:0]	1597	Array
t5_reg[31:0]	0	Array
t6_reg[31:0]	0	Array
P = t/_reg[31:0]	0	Array
▷ = 0 t8_reg[31:0]	0	Array
P = 19_reg[31:0]	0	Array
▷ = v0_reg[31:0]	0	Array
VI_reg[31:0]	0	Arrov
2ero_reg[31	0	Array

## Memory contents in Xilinx after execution

	0	1	2	3	4	5	6	7
0xFF	0	0	0	0	0	0	0	0
0xF7	0	0	0	0	0	0	0	0
0xEF	0	0	0	0	0	0	0	0
0xE7	0	0	0	0	0	0	0	0
0xDF	0	0	0	0	0	0	0	0
0xD7	0	0	0	0	0	0	0	0
0xCF	0	0	0	0	0	0	0	0
0xC7	0	0	0	0	0	0	0	0
0xBF	0	0	0	0	0	0	0	0
0xB7	0	0	0	0	0	0	0	0
0xAF	0	0	0	0	0	0	0	0
0xA7	0	0	0	0	0	0	0	0
0x9F	0	0	0	0	0	0	0	0
0x97	0	0	0	0	0	0	0	0
0x8F	0	0	0	0	0	0	0	0
0x87	0	0	0	0	0	0	0	0
0x7F	0	0	0	0	0	0	0	0
0x77	0	0	0	0	0	0	0	0
0x6F	0	0	0	0	0	0	0	0
0x67	0	0	0	0	0	0	0	0
0x5F	0	0	0	0	0	0	0	0
0x57	0	0	0	0	0	0	0	0
0x4F	0	0	0	0	0	0	0	0
0x47	0	0	0	0	0	0	0	0
0x3F	0	0	0	0	0	0	0	0
0x37	0	0	0	0	0	0	0	0
0x2F	0	0	0	0	0	0	0	0
0x27	0	0	0	0	0	0	0	0
0x1F	0	U	0	0	0	0	0	0
0217	0	0	0	0	0	0	0	1597
0xF	987	610	377	233	144	89	55	34
0x7	21	13	8	5	3	2	1	1

#### Summary

As you can see in our results, both MARS and Xilinx generated the same Fibannoci sequence up to element 17. That means iterative algorithms are working in our single cycle processor implementation. Only problem we encountered was, that the simulation by default runs for 1  $\mu$ S, but to generate 17 elements it needs 1.375  $\mu$ S. Other than that we didn't have any problems at all.

# Conclusion

We successfully managed to implement a working Single cycle processor. For us to be able to accomplish this task we learned a lot of very useful information about how the CPU works. Now we have much better understanding how the CPU works which includes: register file, instruction fetcher, memory, control unit and ALU. We still can further improve this implementation with pipelining and multiple cycles. Current design nowadays is considered to be very slow, since the clock cycle can't be faster than the slowest instruction, which is SW instruction for our case. Overall, we learned a lot from this coursework and we feel satisfied with what we can now do, and we hope to further improve our understanding how the CPU works with multiple cores.