# CS223- Hardware Laboratory

## CS223-MINIPROJECT 3

MINI PROJECT - 3

PRESENTED BY: ANUPAM DAS

1401CS52

### Testing Code

	Machine	Address	Assembly	
main:	20020005	0	addi \$2, \$0, 5	
	2003000C	4	addi \$3, \$0, 12	
	2067FFF7	8	addi \$7, \$3, -9	
	00E22025	С	or \$4, \$7, \$2	
	00642824	10	and \$5, \$3, \$4	
	00A42820	14	add \$5, \$5, \$4	
	10A7000A	18	beq \$5, \$7, end	
	0064202A	1c	slt \$4, \$3, \$4	
	10800001	20	beq \$4, \$0, around	
	20050000	24	addi \$5, \$0, 0	
around:	00E2202A	28	slt \$4, \$7,\$2	
	00853820	2c	add \$7, \$4,\$5	
	00E23822	30	sub \$7, \$7, \$2	
	00E23827	34	nor \$7, \$7, \$2	(Added new)
	00E23826	38	xor \$7, \$7, \$2	(Added new)
	00E53827	3C	nor \$7, \$7, \$5	(Added new)
	20E70005	40	addi \$7, \$7, 5	(Added new)
	30E70005	44	andi \$7, \$7, 5	(Added new)
	00E2381A	48	div \$7, \$7, \$2	(Added new)
	AC670044	4C	sw \$7, 68(\$2)	
	8C020050	50	lw \$2, 80(\$0)	
	08000017	54	j end	(Updated)
	20020001	58	addi \$2, \$0, 1	
end:	AC020054	5C	sw \$2, 84(\$0)	

### Instructions added to extend the single cycle CPU architecture:

- XOR
- DIV
- NOR
- ANDI

#### XOR:

R-Type Instruction.
Opcode: 100110

Changes:

**In aludec** (input [5:0] funct, input [1:0] aluop, output reg [3:0] alucontrol);

Added a case for a new function: 6'b100110: alucontrol <= 4'b0101; // XOR \*

In alu (input [31:0] a, b, input [3:0] alucont, output reg [31:0] result, output zero);

Added a ALU control case

3'b101: result <= a ^ b; //xor \*

#### DIV:

R-Type Instruction. Opcode: 011010

Changes:

**In aludec** (input [5:0] funct, input [1:0] aluop, output reg [3:0] alucontrol);

Added a case for a new function: 6'b011010: alucontrol <= 4'b0100; // DIV \*

In alu (input [31:0] a, b, input [3:0] alucont, output reg [31:0] result, output zero);

Added a ALU control case

3'b100: result <= a / b; //div \*

```
NOR:
```

R-Type Instruction. Opcode: 100111

Changes:

**In aludec** (input [5:0] funct, input [1:0] aluop, output reg [3:0] alucontrol);

Added a case for a new function: 6'b100111: alucontrol <= 4'b0110; // NOR \*

In alu (input [31:0] a, b, input [3:0] alucont, output reg [31:0] result, output zero);

Added a ALU control case

3'b110: result <= ~OR; //nor \*

#### ANDI:

I-Type Instruction. Opcode: 100110

Changes:

**In aludec** (input [5:0] funct, input [1:0] aluop, output reg [3:0] alucontrol);

Utilized spare 2'b11 aluopcode :: 2'b11: alucontrol <= 4'b0000; // andi \*

### Wave of New Updated Instructions used by MIPS single-cycle.

1. 00E23827 34 nor \$7, \$7, \$2 (Added new)

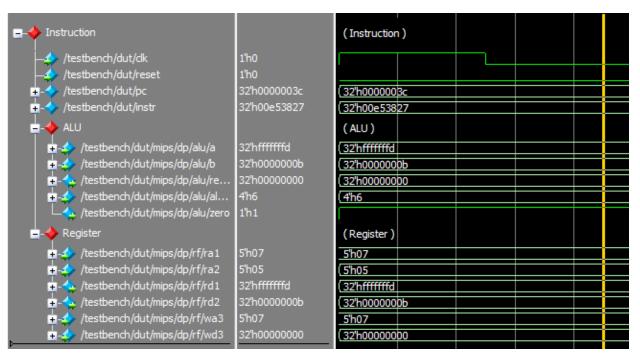
•	, , , ,	,
□-→ Instruction		(Instruction)
/testbench/dut/dk	1'h0	
	1'h0	'
+- / /testbench/dut/pc	32'h00000034	(32'h00000034
I / / / / / / / / / / / / / / / / / / /	32'h00e23827	
T	321100623027	(32'h00e23827
i= ALU		(ALU)
+-4> /testbench/dut/mips/dp/alu/a	32'h00000007	(32h00000007
testbench/dut/mips/dp/alu/b	32'h00000005	32'h00000005
/testbench/dut/mips/dp/alu/re	32'hfffffff8	(32'hffffff8
+-4 /testbench/dut/mips/dp/alu/al	4'h6	(4'h6
/testbench/dut/mips/dp/alu/zero	1'h0	1110
	2110	
E- Register		(Register)
🛓 📣 /testbench/dut/mips/dp/rf/ra1	5'h07	5'h07
±-4/ /testbench/dut/mips/dp/rf/ra2	5'h02	5'h02
+-4 /testbench/dut/mips/dp/rf/rd1	32'h00000007	(32'h00000007
+-4 /testbench/dut/mips/dp/rf/rd2	32'h00000005	32'h00000005
/testbench/dut/mips/dp/rf/wa3	5'h07	5'h07
+-4 /testbench/dut/mips/dp/rf/wd3	32'hfffffff8	(32'hffffff8
- The state of the		(SZIMIMIO

2. 00E23826 38 xor \$7, \$7, \$2 (Added new)

□-→ Instruction		(Instruction)
/testbench/dut/clk	1'h0	
/testbench/dut/reset	1'h0	
+/-/testbench/dut/pc	32'h00000038	32'h00000038
+/- /testbench/dut/instr	32'h00e23826	32'h00e23826
→ ALU		(ALU)
±-4 /testbench/dut/mips/dp/alu/a	32'hfffffff8	32'hffffff8
±-4 /testbench/dut/mips/dp/alu/b	32'h00000005	32'h00000005
±-4 /testbench/dut/mips/dp/alu/re	32'hfffffffd	32'hffffffd
+-4 /testbench/dut/mips/dp/alu/al	4'h5	4'h5
	1'h0	
<b>=</b> - <b>♦</b> Register		(Register)
🛓 📣 /testbench/dut/mips/dp/rf/ra1	5'h07	5'h07
🛓 📣 /testbench/dut/mips/dp/rf/ra2	5'h02	5'h02
🛨 -🗽 /testbench/dut/mips/dp/rf/rd1	32'hfffffff8	32'hffffff8
🚣 -🗽 /testbench/dut/mips/dp/rf/rd2	32'h00000005	32'h00000005
±-≰ /testbench/dut/mips/dp/rf/wa3	5'h07	5'h07
/testbench/dut/mips/dp/rf/wd3	32'hffffffd	32'hffffffd

3.

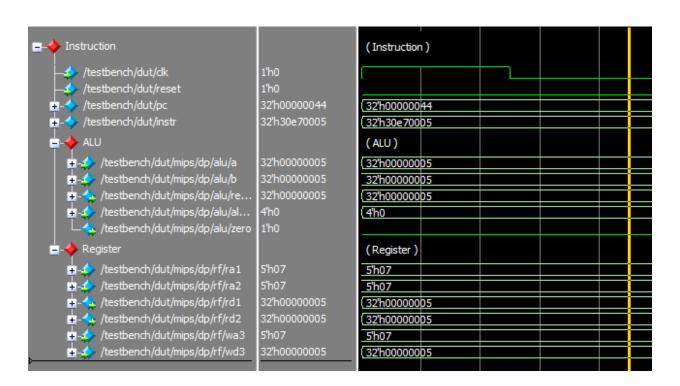
00E53827 3C nor \$7, \$7, \$5 (Added new)



4. 20E70005 40 addi \$7, \$7, 5 (Added new)

■→ Instruction		(Instruction)
/testbench/dut/dk	1'h0	
/testbench/dut/reset	1'h0	
+/ /testbench/dut/pc	32'h00000040	(32'h00000040
testbench/dut/instr	32'h20e70005	(32h20e70005
□		(ALU)
T Y		
testbench/dut/mips/dp/alu/a	32'h00000000	(32'h0000000
testbench/dut/mips/dp/alu/b	32'h00000005	(32'h00000005
testbench/dut/mips/dp/alu/re	32'h00000005	(32'h00000005
testbench/dut/mips/dp/alu/al	4'h2	(4h2
└── /testbench/dut/mips/dp/alu/zero	1'h0	
📥 🔷 Register		(Register)
+ / /testbench/dut/mips/dp/rf/ra1	5'h07	5'h07
testbench/dut/mips/dp/rf/ra2	5'h07	(5'h07
	32'h00000000	(32h00000000
🗓 🔩 /testbench/dut/mips/dp/rf/rd2	32'h00000000	(32h00000000
testbench/dut/mips/dp/rf/wa3	5'h07	(5'h07
	32'h00000005	(32h00000005
<del>- "</del>		

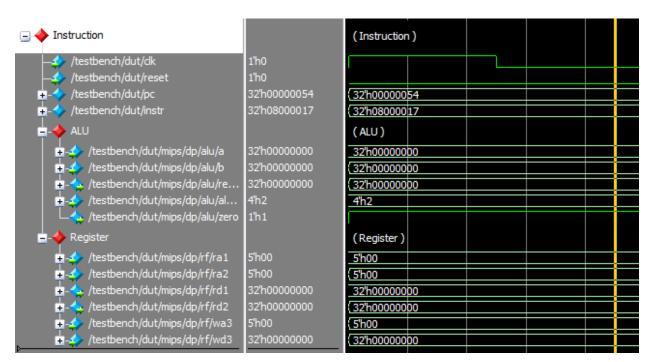
5. 30E70005 44 andi \$7, \$7, 5 (Added new)



6. 00E2381A 48 div \$7, \$7, \$2 (Added new)

■-  Instruction		(Instruction)
/testbench/dut/dk	1'h0 1'h0	
/testbench/dut/reset ///testbench/dut/pc	32'h00000048	(32'h00000048
i	32'h00e2381a	(32'h00e238i1a (ALU)
testbench/dut/mips/dp/alu/a	32'h00000005 32'h00000005	32'h00000005 (32'h00000005
testbench/dut/mips/dp/alu/re testbench/dut/mips/dp/alu/re	32'h00000001 4'h4	(32'h00000001 (4'h4
/testbench/dut/mips/dp/alu/zero	1'h0	(Register)
/testbench/dut/mips/dp/rf/ra1	5'h07 5'h02	5'h07
testbench/dut/mips/dp/rf/ra2 testbench/dut/mips/dp/rf/rd1	32'h00000005 32'h00000005	(5'h02 32'h00000005
testbench/dut/mips/dp/rf/rd2 t-4 /testbench/dut/mips/dp/rf/wa3	5'h07	32'h00000005 (5'h07
±-�/testbench/dut/mips/dp/rf/wd3	32'h00000001	(32'h00000001

7. 08000017 54 j end (Updated)



8. end: AC020054 5C sw \$2, 84(\$0)

■  Instruction		(Instruction)
/testbench/dut/clk	1'h0	
/testbench/dut/reset	1'h0	
+- / /testbench/dut/pc	32'h0000005c	(32'h0000005c
+- / /testbench/dut/instr	32'hac020054	(32'hac020054
T (	32.113232333	
Ē- <b>→</b> ALU		(ALU)
i de de la destación de la definition d	32'h00000000	32'h00000000
±-≰ /testbench/dut/mips/dp/alu/b	32'h00000054	(32'h00000054
±-4 /testbench/dut/mips/dp/alu/re	32'h00000054	(32'h00000054
testbench/dut/mips/dp/alu/al	4'h2	4'h2
/testbench/dut/mips/dp/alu/zero	1'h0	
		(Register)
testbench/dut/mips/dp/rf/ra1	5'h00	5'h00
testbench/dut/mips/dp/rf/ra2	5'h02	(5h02
testbench/dut/mips/dp/rf/rd1	32'h00000000	32'h0000000
	32'h00000001	(32'h00000001
testbench/dut/mips/dp/rf/wa3	5'h02	(5'h02
I A	32'h00000054	
+-4/ /testbench/dut/mips/dp/rf/wd3	321100000034	(32'h00000054

THANK YOU.