CSE332

Computer Organization And Architecture

10-bit Custom RISC-V Microprocessor

Submitted By:
Sheikh Tanzida Tasmim Erin
ID:2013681042

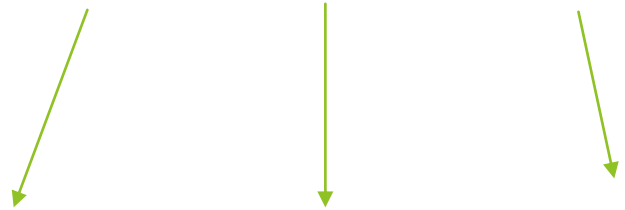
Control Signals

	opcode	reg_write	reg_imm	beq/bne	beq con	mem _enb	lw	sw	mem/reg
add	0000	1	0	0	0	0	0	0	0
sub	0001	1	0	0	0	0	0	0	0
lw	0010	1	1	0	0	1	1	0	1
sw	0011	1	1	0	0	1	0	1	1
and	0100	1	0	0	0	0	0	0	0
andi	0101	0	1	0	0	0	0	0	0
or	0110	1	0	0	0	0	0	0	0
ori	0111	0	1	0	0	0	0	0	0
beq	1000	0	1	1	1	0	0	0	0
bne	1001	0	1	1	1	0	0	0	0

sw $3, 1($2)

opcode(4 bit)	rd(2 bit)	rs(2 bit)	imm(2 bit)
sw	$3	$2	1
0011	11	10	01

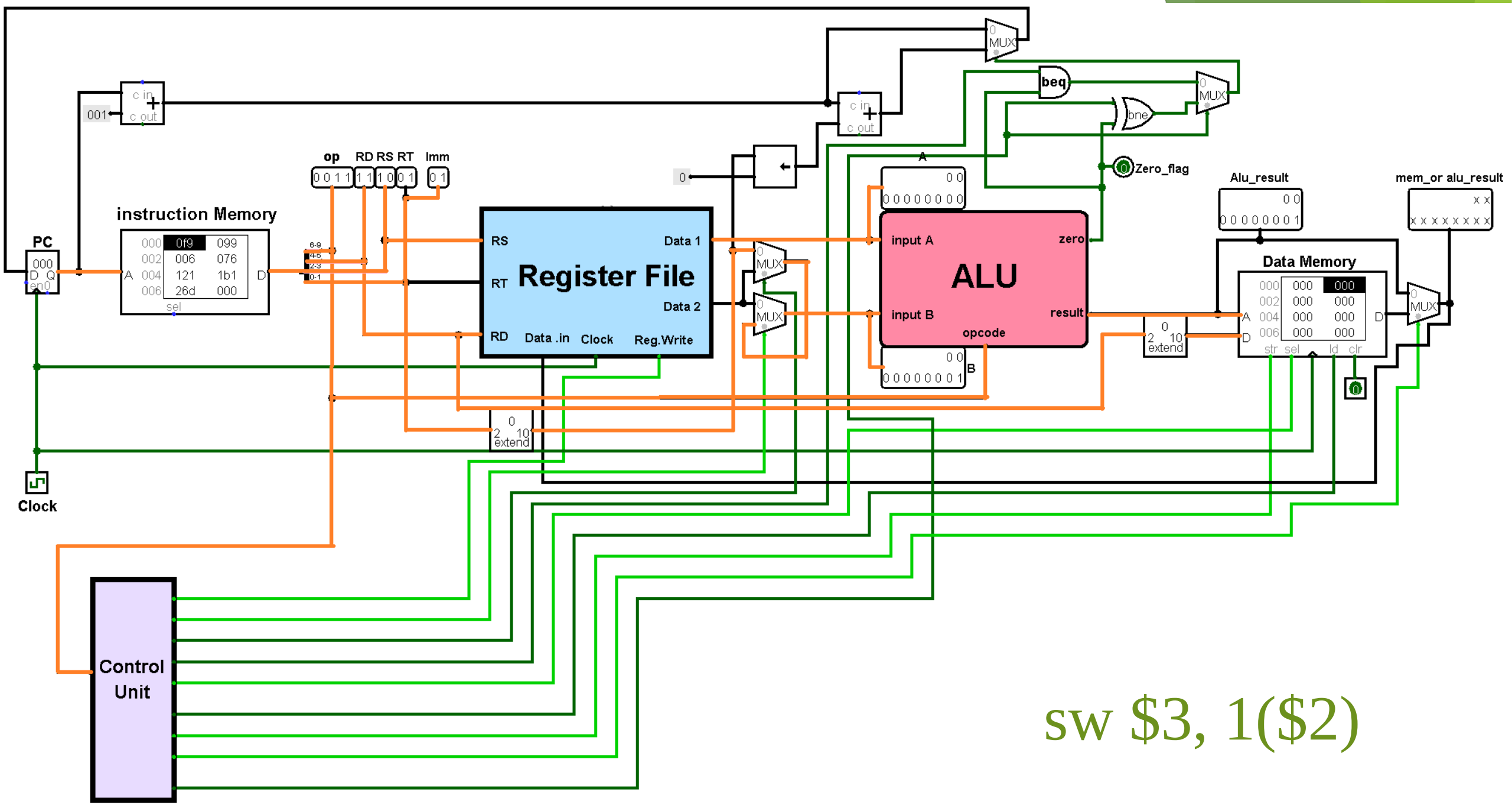
Machine Code: 0000 1111 1001



Hexadecimal: 0 f 9

sw $3, 1($2)

In store word, the value in $2 is added with offset 1. The result obtained is an address. The value in $3 is stored in that address.

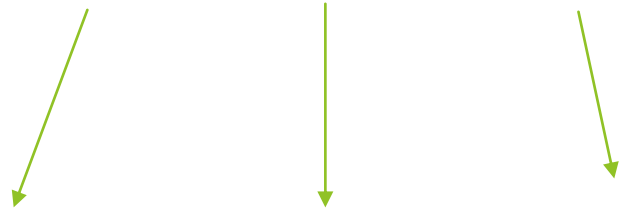

c in
c out
001
op
RD RS RT
Imm
0 0 1 1 1 1 1 0 0 1
0 1
instruction Memory
PC
000
D Q
en0
A
D
sel
000 0f9 099
002 006 076
004 121 1b1
006 26d 000
6-9
4-5
2-3
0-1
RS
RT
RD
Data .in
Clock
Reg.Write
Register File
Data 1
Data 2
0
2 10
extend
MUX
beq
bne
Zero_flag
A
0 0
0 0 0 0 0 0 0 0
input A
input B
opcode
zero
result
ALU
B
0 0
0 0 0 0 0 0 0 1
Alu_result
mem_or alu_result
x x
x x x x x x x x
Data Memory
000 000 000
002 000 000
004 000 000
006 000 000
str sel ld clr
Clock
Control
Unit

sw $3, 1($2)

lw $1, 1($2)

opcode(4 bit)	rd(2 bit)	rs(2 bit)	imm(2 bit)
lw	$1	$2	1
0010	01	10	01

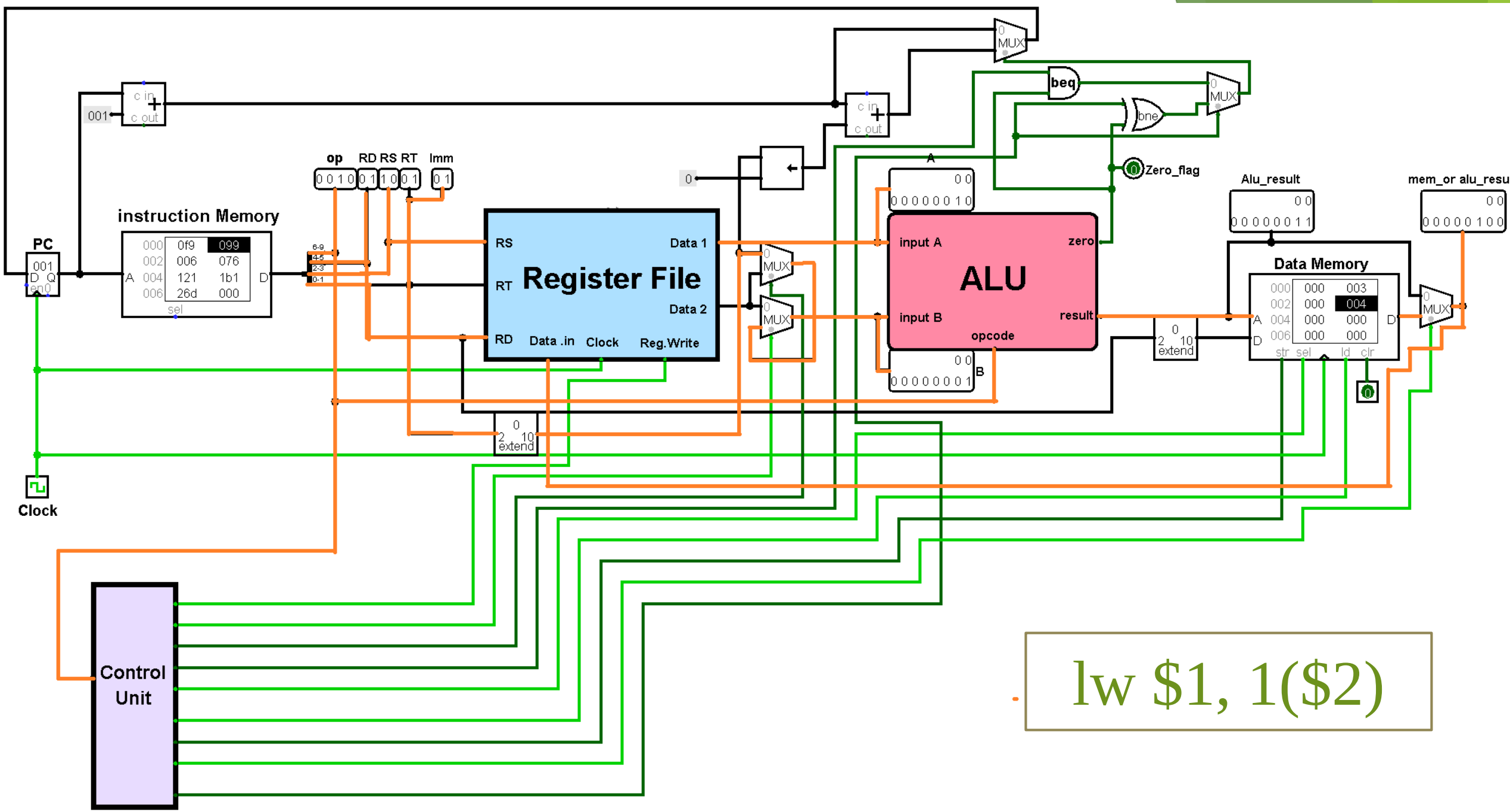
Machine Code: 0000 1001 1001



Hexadecimal: 0 9 9

lw $1, 1($2)

In load word, the value in $2 is added with offset 1. The result obtained is an address. The value in that address is loaded in $1 register.

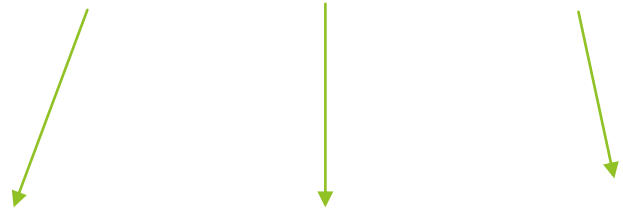

c in
c out
001
op
RD RS RT
Imm
instruction Memory
PC
sel
RS
RT
RD
Register File
Data 1
Data 2
Data .in
Clock
Reg.Write
MUX
beq
bne
Zero_flag
A
input A
input B
ALU
zero
result
opcode
B
extend
Alu_result
mem_or alu_result
Data Memory
str
sel
ld
clr
Clock
Control Unit

lw $1, 1($2)

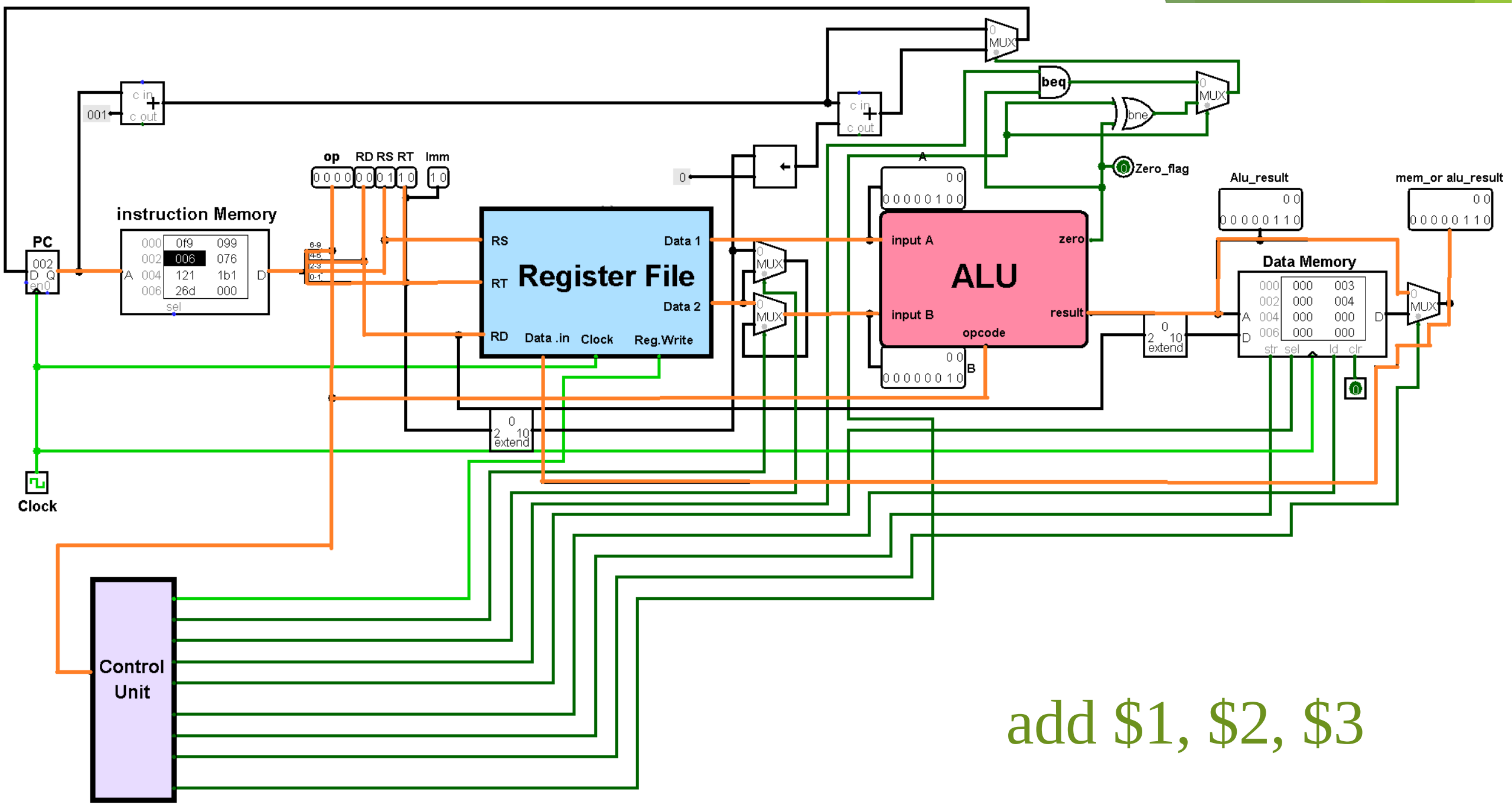
add $0, $1, $2

opcode(4 bit)	rd(2 bit)	rs(2 bit)	rt(2 bit)
add	$0	$1	$2
0000	00	01	10

Machine Code: 0000 0000 0110



Hexadecimal: 0 0 6


c in
c out
001
op
RD RS RT
Imm
instruction Memory
PC
sel
Register File
RS
RT
RD
Data 1
Data 2
Data .in
Clock
Reg.Write
MUX
beq
bne
Zero_flag
A
ALU
input A
input B
opcode
zero
result
B
extend
Alu_result
Data Memory
str sel ld clr
mem_or alu_result
Clock
Control Unit

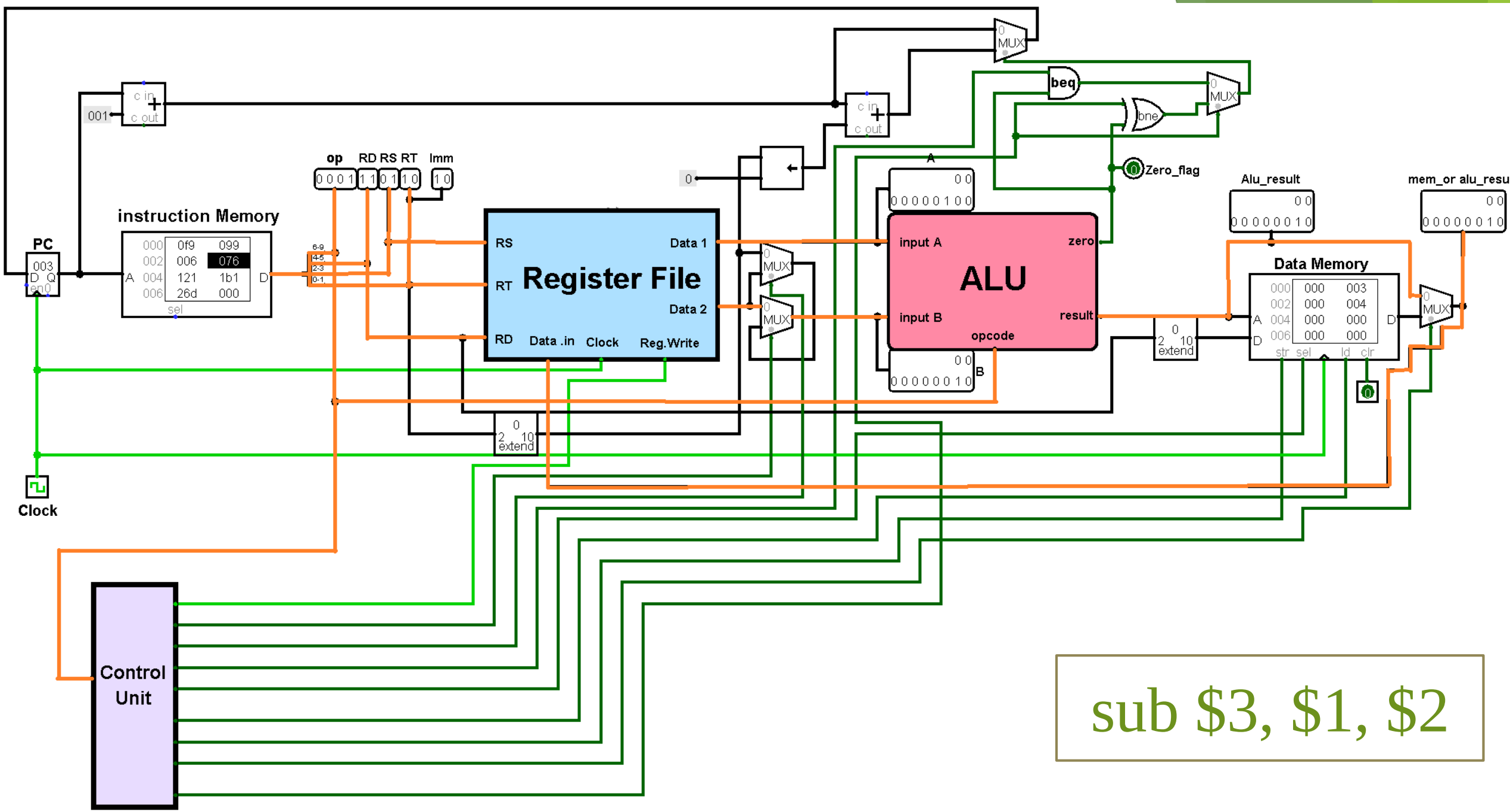
add $1, $2, $3

sub $3, $1, $2

opcode(4 bit)	rd(2 bit)	rs(2 bit)	rt(2 bit)
sub	$3	$1	$2
0001	11	01	10

Machine Code: 0000 0111 0110

Hexadecimal: 0 7 6

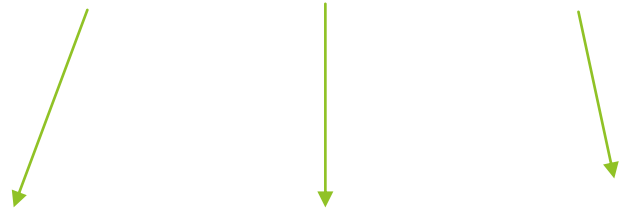

c in
c out
001
op
RD RS RT
Imm
instruction Memory
PC
sel
RS
RT
RD
Data 1
Data 2
Register File
Data .in
Clock
Reg.Write
MUX
beq
bne
Zero_flag
A
B
input A
input B
opcode
zero
result
ALU
extend
Alu_result
Data Memory
str sel ld clr
mem_or alu_result
Clock
Control Unit

sub $3, $1, $2

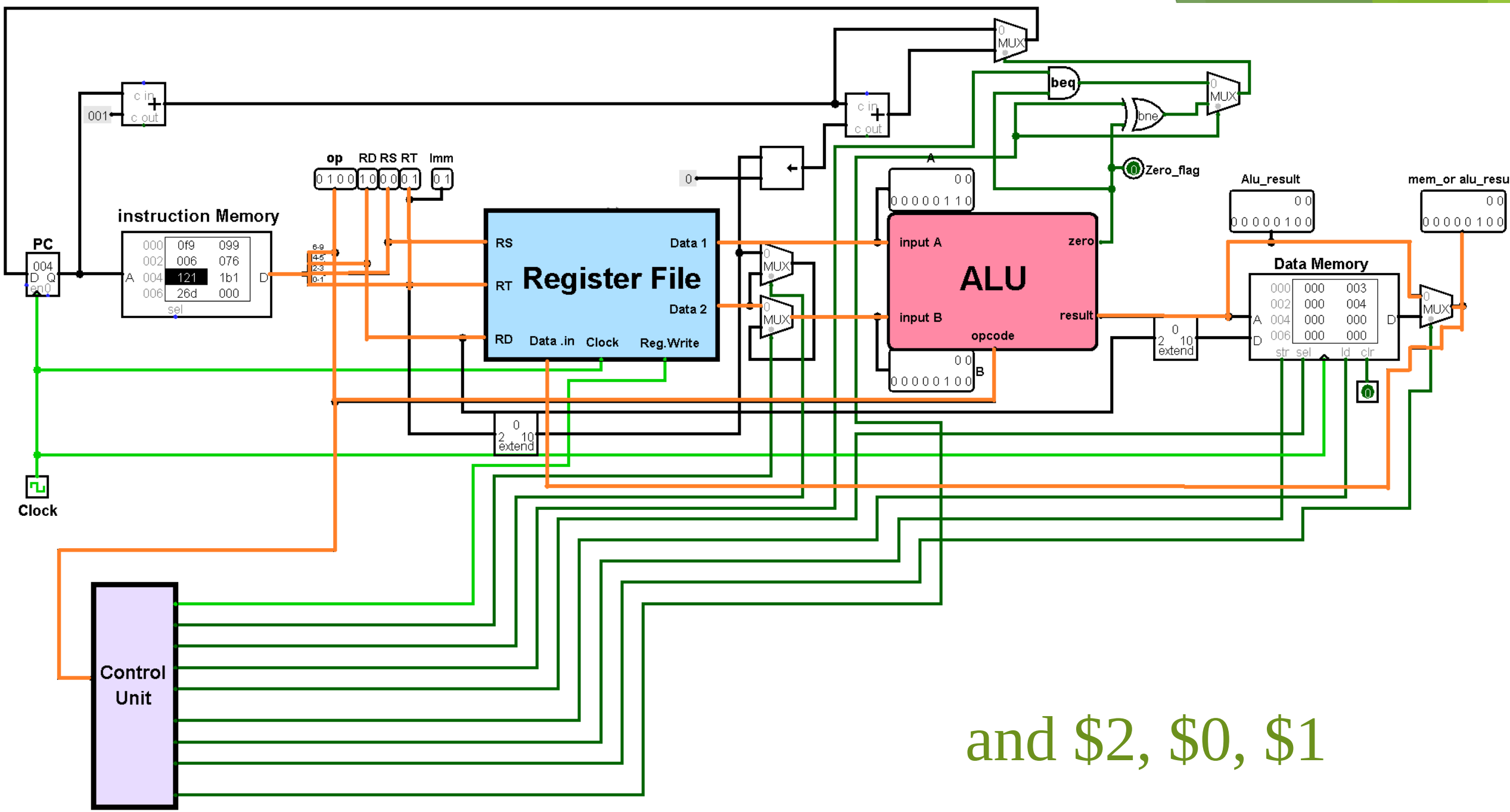
and $2, $0, $1

opcode(4 bit)	rd(2 bit)	rs(2 bit)	rt(2 bit)
and	$2	$0	$1
0100	10	00	01

Machine Code: 0001 0010 0001



Hexadecimal: 1 2 1

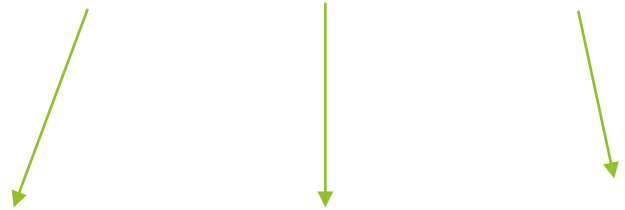

c in
c out
001
op RD RS RT Imm
0 1 0 0 1 0 0 0 0 1 0 1
instruction Memory
PC
004
clr
sel
000 0f9 099
002 006 076
A 004 121 1b1 D
006 26d 000
6-9
4-5
2-3
0-1
RS
RT
RD
Data .in
Clock
Reg.Write
Register File
Data 1
Data 2
0
2 10
extend
MUX
beq
bne
Zero_flag
A
0 0
0 0 0 0 0 1 1 0
input A
input B
opcode
zero
result
ALU
0 0
0 0 0 0 0 1 0 0
B
Alu_result
mem_or alu_result
Data Memory
000 000 003
002 000 004
A 004 000 000 D
006 000 000
str sel ld clr
Clock
Control
Unit

and $2, $0, $1

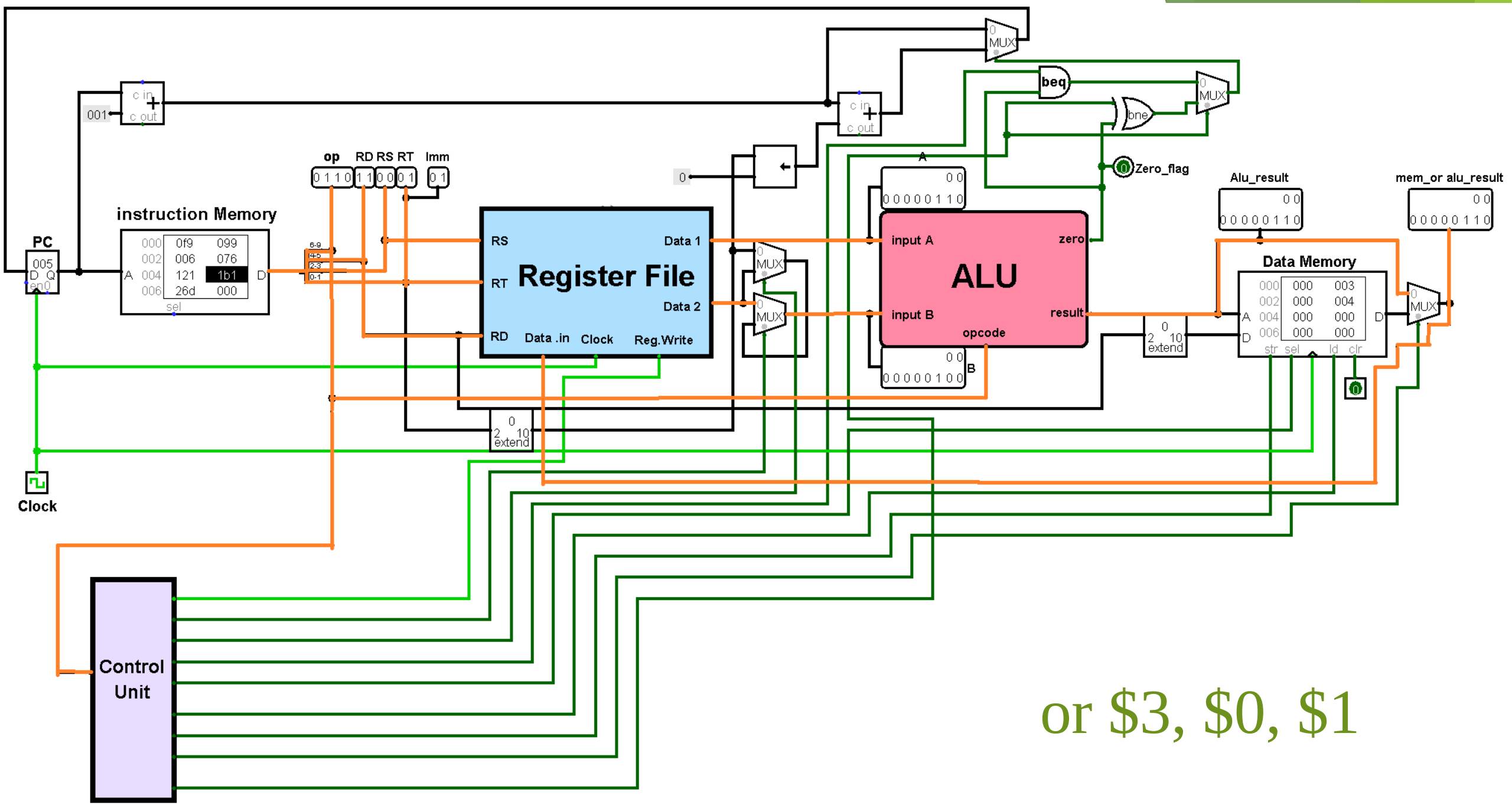
or $3, $0, $1

opcode(4 bit)	rd(2 bit)	rs(2 bit)	rt(2 bit)
or	$3	$0	$1
0110	11	00	01

Machine Code: 0001 1011 0001



Hexadecimal: 1 b 1

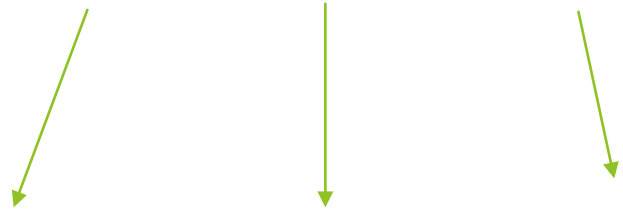

c in
c out
001
op RD RS RT Imm
0 1 1 0 1 1 0 0 0 1 0 1
instruction Memory
PC
005
clk
A
D
sel
000 0f9 099
002 006 076
004 121 1b1
006 26d 000
6-9
4-5
2-3
0-1
RS
RT
RD
Data 1
Data 2
Register File
Data .in
Clock
Reg.Write
0
2 10
extend
MUX
beq
bne
Zero_flag
A
0 0
0 0 0 0 0 1 1 0
input A
input B
ALU
zero
result
opcode
0 0
0 0 0 0 0 1 0 0
B
Alu_result
mem_or alu_result
Data Memory
000 000 003
002 000 004
004 000 000
006 000 000
str sel ld clr
Clock
Control
Unit

or $3, $0, $1

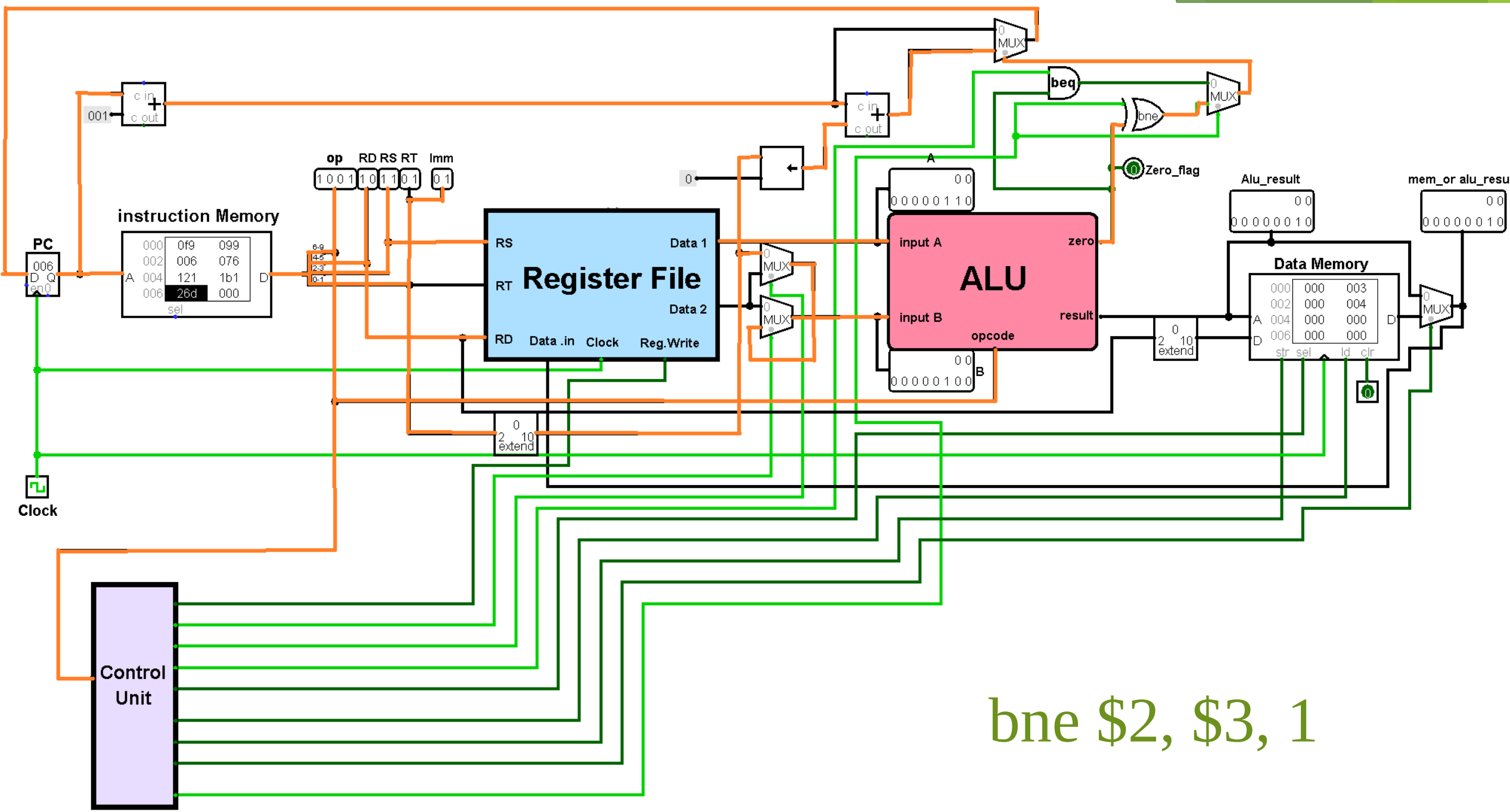
bne $2, $3, 1

opcode(4 bit)	rd(2 bit)	rs(2 bit)	imm(2 bit)
bne	$2	$3	1
1001	10	11	01

Machine Code: 0010 0110 1101



Hexadecimal: 2 6 d

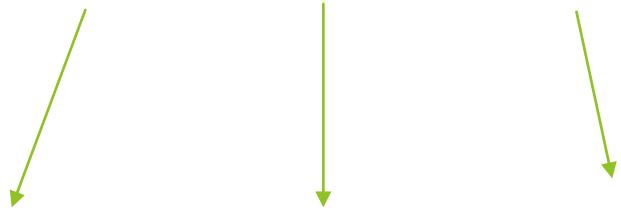

c in
c out
001
op
RD RS RT
Imm
1 0 0 1
1 0
1 1
0 1
0 1
instruction Memory
PC
006
D Q
en0
A
D
000 0f9 099
002 006 076
004 121 1b1
006 26d 000
sel
6-9
4-5
2-3
0-1
RS
RT
RD
Data 1
Data 2
Register File
Data .in
Clock
Reg.Write
0
2 10
extend
MUX
beq
bne
Zero_flag
A
0 0
0 0 0 0 0 1 1 0
input A
input B
opcode
zero
result
ALU
0 0
0 0 0 0 0 1 0 0
B
Alu_result
0 0
0 0 0 0 0 0 1 0
mem_or alu_result
0 0
0 0 0 0 0 0 1 0
Data Memory
000 000 003
002 000 004
004 000 000
006 000 000
A
D
str sel ld clr
Clock
Control
Unit

bne $2, $3, 1

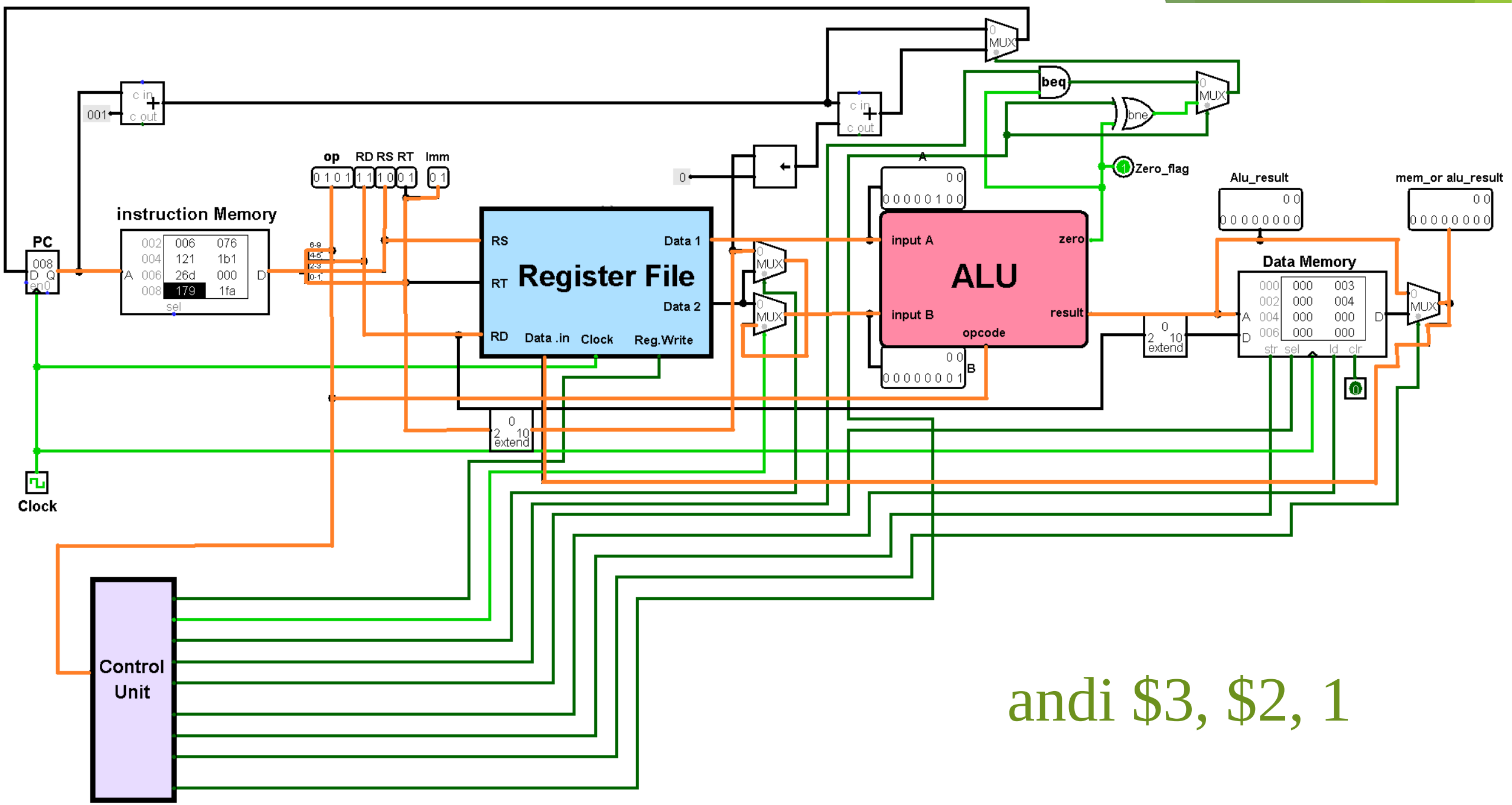
andi $3, $2, 1

opcode(4 bit)	rd(2 bit)	rs(2 bit)	imm(2 bit)
andi	$3	$2	1
0101	11	10	01

Machine Code: 0001 0111 1001



Hexadecimal: 1 7 9

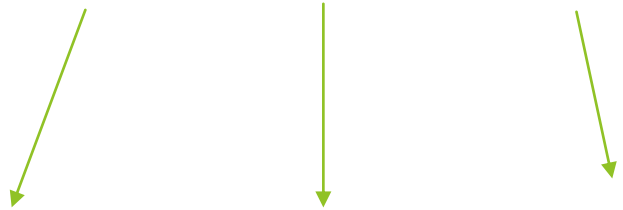

c in
c out
001
op RD RS RT Imm
0 1 0 1 1 1 1 0 0 1 0 1
instruction Memory
PC
002 006 076
004 121 1b1
006 26d 000
008 179 1fa
sel
RS
RT
RD
Data 1
Data 2
Data .in
Clock
Reg.Write
Register File
MUX
beq
bne
Zero_flag
A
input A
input B
opcode
zero
result
ALU
B
extend
Alu_result
mem_or alu_result
Data Memory
000 000 003
002 000 004
004 000 000
006 000 000
str sel ld clr
Clock
Control Unit

andi $3, $2, 1

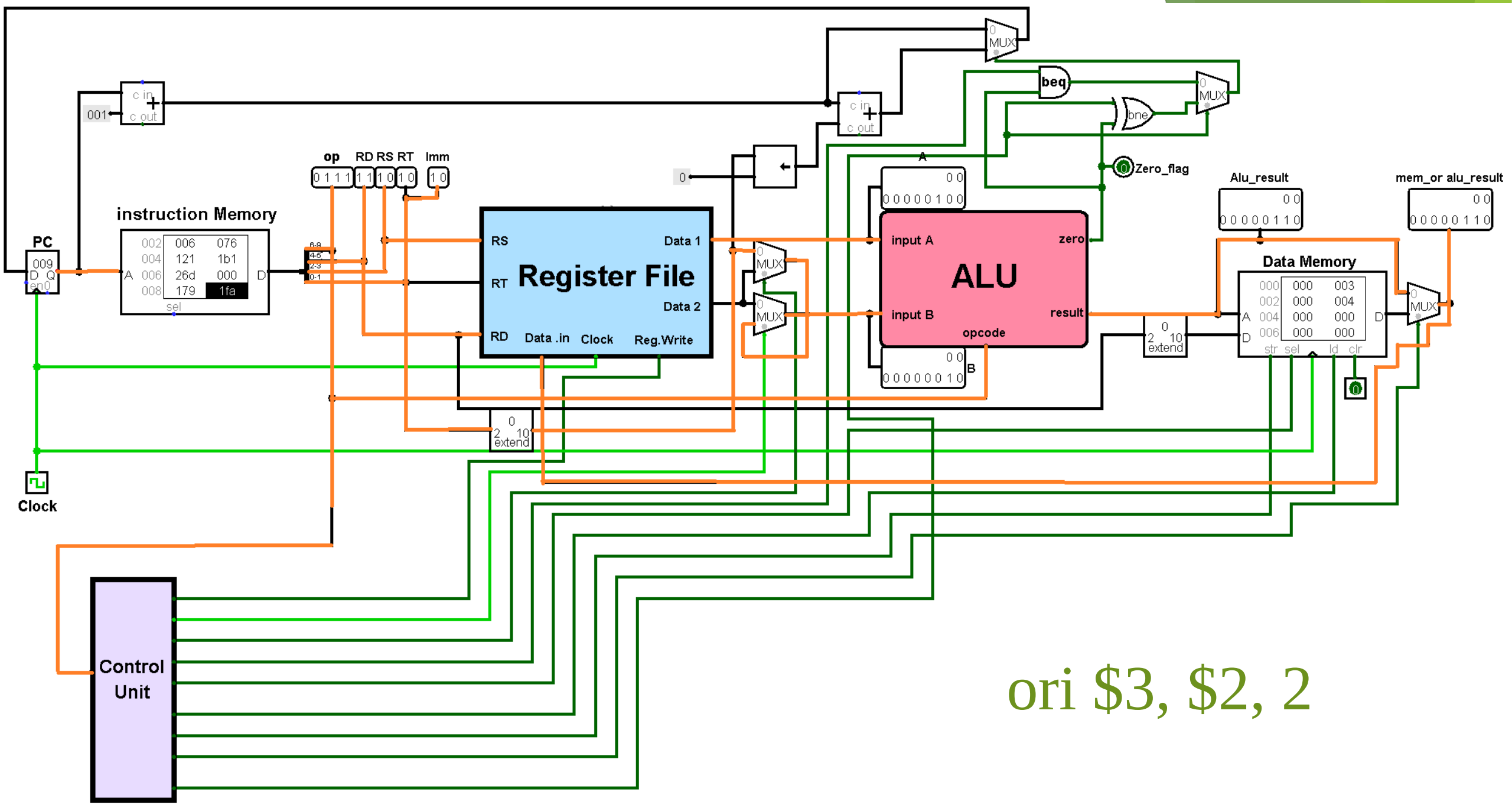
ori $3, $2, 2

opcode(4 bit)	rd(2 bit)	rs(2 bit)	imm(2 bit)
ori	$3	$2	2
0111	11	10	10

Machine Code: 0001 1111 1010



Hexadecimal: 1 f a


c in
c out
001
op RD RS RT Imm
0 1 1 1 1 1 1 0 1 0 1 0
instruction Memory
PC
009
D Q
en0
A
002 006 076
004 121 1b1
006 26d 000
008 179 1fa
D
sel
RS
RT
RD
Register File
Data 1
Data 2
Data .in
Clock
Reg.Write
0
2 10
extend
MUX
beq
bne
Zero_flag
A
0 0
0 0 0 0 0 1 0 0
input A
input B
ALU
zero
result
opcode
0 0
0 0 0 0 0 0 1 0
B
Alu_result
0 0
0 0 0 0 0 1 1 0
mem_or alu_result
0 0
0 0 0 0 0 1 1 0
Data Memory
000 000 003
002 000 004
004 000 000
006 000 000
A
D
str sel ld clr
Clock
Control
Unit

ori $3, $2, 2