

## **Lab 5 - Building a Simple Processor**

### 1. Instruction Encoding and assigning OP-Codes

- Control signals have been directly added to the op codes using method below.

Instruction	OPCODE					
	Shift_Left [7]	BranchEqual [6]	Write_Enable [5]	2'sCmp Enable [4]	Alu Src [3]	AluOP [2:0]
loadi	0	0	1	0	0	000
mov	0	0	1	0	1	000
Add	0	0	1	0	1	001
sub	0	0	1	1	1	001
and	0	0	1	0	1	010
or	0	0	1	0	1	011
beq	0	1	0	1	1	001
j	0	0	0	0	0	000
mul	0	0	1	0	1	100
srl	0	0	1	0	1	101
sll	1	0	1	0	1	101
sra	0	0	1	0	1	110
ror	0	0	1	0	1	111
bne	0	0	0	1	0	001

- Two additional control signals were added manually since the lack of bit positions
  - ◆ Jump Control Signal
    - This control signal is 1 only in the jump instruction , and 0 otherwise.
  - ◆ Branch Not equal Control Signal
    - This control signal is 1 only in the bne instruction , and 0 otherwise.

## Final Opcodes

Instruction	Opcode
loadi	00100000
mov	00101001
add	00101001
sub	00111001
and	00101010
or	00101011
mul	00101100
srl	00100101
sll	10100101
sra	00100110
ror	00100111
beq	01011001
bne	00011001
j	00000000

## 2. Timing Details

ALU OP	Function	Instruction	Latency
000	forward	loadi, mov	#1
001	add	add,sub	#2
010	and	and	#1
011	or	or	#1
100	multiplication	mul	#2
101	Shift logical	sll, srl	#2
110	Shift arithmetic	sra	#2
111	Rotate	ror	#2

### 3. Final Data-paths

