Control

7	**6**	**5**	**4**			instr[7:4]						
	opcode		funct	instruction	type	control receives	jctrl	jrctrl	memwrite	memread	memtoreg	ALUsrc
0	0	0	0	add	R	0000	0	0	0	0	0	0
0	0	0	1	nop	n/a	0001	0	0	0	0	0	0
0	0	1	0	nand	R	0010	0	0	0	0	0	0
0	0	1	1	nop	n/a	0011	0	0	0	0	0	0
0	1	0	0	slt_0	R	0100	0	0	0	0	0	0
0	1	0	1	slt_1	R	0101	0	0	0	0	0	0
0	1	1	0	sl	R	0110	0	0	0	0	0	0
0	1	1	1	sr	R	0111	0	0	0	0	0	0
1	0	0	0	lw	I	1000	0	0	0	1	1	1
1	0	0	1	sw	I	1001	0	0	1	0	0	1
1	0	1	0	addi	I	1010	0	0	0	0	0	1
1	0	1	1	jr	JR	1011	0	1	0	0	0	0
1	1	0	n/a	beq	J	1100	1	0	0	0	0	0
1	1	0	0	nop	n/a	1100	0	0	0	0	0	0
1	1	1	n/a	jal	J	1110	1	0	0	0	0	0
1	1	1	1	nop	n/a	1111	0	0	0	0	0	0

Note

ALU	at 0, returns 1 if inputs are equal (subtract) at 'result', returns the calculated output
Jump Instruction	the immediate value is the amount jumped from PC (PC relative)
$ra	$ra holds the fixed memory address 10110011 (8-bit) which is addressed directly by JR type

Team Aardvark
ECE151A Sp16

regwrite	ALUop	Ctrl from ALUctrl
1	111	1
0	000	0
1	001	1
0	000	0
1	010	1
1	010	1
1	011	1
1	100	1
1	000	0
1	000	0
1	111	1
0	000	0
0	000	0
0	000	0
0	000	0
0	000	0