Instruction Format:

16 bit instructions

Name	Bit fields					Notes
	4 Bits	3 Bits	3 Bits	3 Bits	3 Bits	
R - Format	Op	rs	rt	rd	funct	Arithmetic, Logic instruction
I - Format	Op	rs	rt	Immediate (6 Bits)		Load, Store, Branch, Immediate
J - Format	Op	Target Address (12 bits)				Jump

OP Code and func bits mapping:

Op Code	Instruction Format	ALU Op	func	Instruction Operation	ALU Operation	ALU Control
1001	I-Type	00	XXX	sw (Load Word)	ADD	010
1010	I-Type	00	XXX	sw (Store Word)	ADD	010
0101	I-Type	01	XXX	beq (Branch if equal)	SUB	110
0000	R-Type	10	000	ADD	ADD	010
0000	R-Type	10	010	SUB	SUB	110
0000	R-Type	10	100	AND	AND	000
0000	R-Type	10	101	OR	OR	001
0000	R-Type	10	001	MUL	MUL	111
0000	R-Type	10	011	DIV	DIV	011
1101	J-Type	XX	XXX	Jump		

Available Operations (With Example):

Operation	Opcode	funct	Format	Type	Assembly Format	Action
ADD (Addition)	0000	000	R-format	Arithmetic	add $1, $2 ,$3	$1 = $2 + $3
SUB (Subtraction)	0000	010	R-format	Arithmetic	sub $1, $2, $3	$1 = $3 - $2
AND	0000	100	R-format	Logical	and $1, $2, $3	$1 = $2 &$3
OR	0000	101	R-format	Logical	or $1, $2, $3	$1 = $2 | $3
MUL (Multiply)	0000	011	R-format	Arithmetic	mul $1,$2,$3	$1=$2*$3
DIV (Divide)	0000	001	R-format	Arithmetic	div $1,$2,$3	$1=$3/$2
ADDI (Add immediate)	0010	XXX	I-format	Arithmetic	addi $1,$2,100	$1 = $2+100
BEQ (Branch if equal)	0101	XXX	I-format	Conditional Branch	beq $1,$2,100	if($1==$2) go to PC+4+100
LW (Load Word)	1001	XXX	I-format	Data Transfer	lw $1,100($2)	$1 = Memory[$2+100]
SW (Store Word)	1010	XXX	I-format	Data Transfer	sw $1,100($2)	Memory[$2+100] =$1
JUMP	1101	XXX	J-Format	Unconditional Branch	Jump 10	Jump to location 10 of instruction memory