1) Arithmetic instructions

1.1 R-type

OP code	rs(source register)	rt(destination register)	Don't care	func
6 bits	5 bits	5 bits	10 bits	6 bits

Instruction	OP code	func
add rs,rt	000000	000001
Comp rs,rt	000000	000010
and rs,rt	000000	000011
xor rs,rt	000000	000100
diff rs,rt	000000	000101

1.2 I-type

OP code	rs	rs	Immediate value
6 bits	5 bits	5 bits	16 bits

Instruction	OP code
addi rs,imm	000001
compi rs,imm	000010

2) Shift

2.1

OP code	rs(source register)	rt(destination register)	Don’t care	func
6 bits	5 bits	5 bits	10 bits	6 bits

Instruction	OP code	func
shllv rs,rt	000000	000110
shrl rs,rt	000000	000111
shrav rs,rt	000000	001000

2.2

OP code	rs	rs	Immediate value
6 bits	5 bits	5 bits	16 bits

shll rs,sh	000011
shrl rs,sh	000100
shra rs,sh	000101

2) Memory instructions

OP code	rt(destination register)	rs(source register)	Immediate value
6 bits	5 bits	5 bits	16 bits

instructions	OP code
lw rt,imm(rs)	000110
sw rt,imm(rs)	000111

3) Branch Instructions

3.1 Conditional jump

OP code	Don’t care	rs	L
6 bits	5 bits	5 bits	16 bits

instructions	OP code
bltz rs,L	001000
bz rs,L	001001
bnz rs,L	001010

3.2 Unconditional jump

3.2.1 Based on register

OP code	Don’t care	rs(source register)	Don’t care
6 bits	5 bits	5 bits	16 bits

instructions	OP code
br rs	001011

3.2.2 Based on L

OP code	Don’t care	L
6 bits	10 bits	16 bits

instructions	OP code
b, L	001100
bcy, L	001101
bncy, L	001110

3.2.3 Jump and Link (for subroutines)

OP code	Don’t care	L
6 bits	10 bits	16 bits

instructions	OP code
bl L	001111

4) Halt instruction

This instruction stops the program counter from incrementing, and the processor comes to a halt

OP code	Don’t care
111111	26 bits

5) NOP instruction

Does nothing

OP code	Don’t care
111110	26 bits