Definitions

RS = Register[IR[25-21]]
RT = Register[IR[20-16]]
RD = Register[IR[15-11]]

JT = IR[25-0]]
IMM = IR[15-0]

OP = IR[5-0]]
FUNC = IR[31-26]

Common RTL

Cycle	**Label**	**RTL**	**Control**
0	Fetch:	PC <= PC + 4 IR <= Mem[PC]	ALUSelA <= 0 ALUSelB <= 001 MemRead_L <= 0 IorD <= 0 IRWrite_H <= 1 PCWrite_H <= 1 PCSource <= 0
1	Decode:	Target <= PC + SEx(Imm << 2)	ALUSelA <= 0 ALUSelB <= 011 TargetWrite_H <= 1
Last	Done:	goto Fetch	

Memory Read

Cycle	**Label**	**RTL**	**Control**
?	MR:	if MMRESP_H then goto MR_Done	MMREAD_L
?	MR_LOOP:	if MMRESP_H then goto MR_Done else goto MR_Loop	
?	MR_DONE:		

Memory Write

Cycle	**Label**	**RTL**	**Control**
?	MW:	if MMRESP_H then goto MW_Done	MMWRITE_L
?	MW_LOOP:	if MMRESP_H then goto MW_Done else goto MW_Loop	
?	MW_DONE:		

add	Type **R-Type**	Op (31:26) **000000**	Func (5:0) **100000**
Cycle	**Label**	**RTL**	**Control**
2	ADD:	ALUOut <= RS + RT	ALUSelA <= 1 ALUSelB <= 000

3		RD <= ALUOut	ALUSelA <= 1 ALUSelB <= 000 RegDst <= 01 MemToReg <= 00 RegWrite_H <= 1

addi	Type **I-Type**	Op (31:26) **001000**	
Cycle	**Label**	**RTL**	**Control**
2	ADDI:	ALUOut <= RS + SEx(Imm)	ALUSelA <= 1 ALUSelB <= 010
3		RT <= ALUOut	ALUSelA <= 1 ALUSelB <= 010 RegDst <= 01 MemToReg <= 00 RegWrite_H <= 1

and	Type **R-Type**	Op (31:26) **000000**	Func (5:0) **100100**
Cycle	**Label**	**RTL**	**Control**
2	AND:	ALUOut <= RS AND RT	ALUSelA <= 1 ALUSelB <= 000
3		RD <= ALUOut	ALUSelA <= 1 ALUSelB <= 000 RegDst <= 01 MemToReg <= 00 RegWrite_H <= 1

andi	Type **I-Type**	Op (31:26) **001100**	
Cycle	**Label**	**RTL**	**Control**
2	ANDI:	ALUOut <= RS AND ZEx(Imm)	ALUSelA <= 1 ALUSelB <= 010
3		RT <= ALUOut	ALUSelA <= 1 ALUSelB <= 100 RegDst <= 01 MemToReg <= 00 RegWrite_H <= 1

beq	Type **I-Type**	Op (31:26) **001000**	
Cycle	**Label**	**RTL**	**Control**

2	BEQ:	ALUOut <= RS - RT	ALUSelA <= 1 ALUSelB <= 000
3		if (Zero) then PC <= Target	ALUSelA <= 1 ALUSelB <= 000 PCWriteZero_H <= 1 PCSource <= 01

bne	Type **I-Type**	Op (31:26) **001001**	
Cycle	**Label**	**RTL**	**Control**
2	BNE:	ALUOut <= RS - RT	ALUSelA <= 1 ALUSelB <= 000
3		if (!Zero) PC <= Target	ALUSelA <= 1 ALUSelB <= 000 PCWriteNZero_H <= 1 PCSource <= 01

j	Type **J-Type**	Op (31:26) **000010**	
Cycle	**Label**	**RTL**	**Control**
2	J:	PC <= PC[31-28] || (JT << 2)	PCWrite_H <= 1 PCSource <= 10

jal	Type **J-Type**	Op (31:26) **000011**	
Cycle	**Label**	**RTL**	**Control**
2	JAL:	R31 <= PC PC <= PC[31-28] || (JT << 2)	RegDst <= 10 MemToReg <= 10 RegWrite_H <= 1 PCWrite <= 1 PCSource <= 10

jr	Type **R-Type**	Op (31:26) **000000**	Func (5:0) **001000**
Cycle	**Label**	**RTL**	**Control**
2	JR:	PC <= RS	PCWrite <= 1 PCSource <= 11

lui	Type **I-Type**	Op (31:26) **001111**	

Cycle	**Label**	**RTL**	**Control**
2	LUI:	RT <= IMM || 0^{16}	RegDst <= 00 MemToReg <= 11 RegWrite_H <= 1

lw	Type **I-Type**	Op (31:26) **100011**	
Cycle	**Label**	**RTL**	**Control**
2	LW:	ALUOut <= RS + IMM	ALUSelA <= 1 ALUSelB <= 010
3		RT <= Mem[ALUOut]	MemRead_L <= 0 IorD = 1 ALUSelA <= 1 ALUSelB <= 010 RegDst <= 00 MemToReg <= 01 RegWrite_H <= 1

or	Type **R-Type**	Op (31:26) **000000**	Func (5:0) **100101**
Cycle	**Label**	**RTL**	**Control**
2	OR:	ALUOut <= RT OR RS	ALUSelA <= 1 ALUSelB <= 000
3		RD <= ALUOut	ALUSelA <= 1 ALUSelB <= 000 RegDst <= 01 MemToReg <= 00 RegWrite_H <= 1

ori	Type **I-Type**	Op (31:26) **001101**	
Cycle	**Label**	**RTL**	**Control**
2	ORI:	ALUOut <= RT OR ZEx(IMM)	ALUSelA <= 1 ALUSelB <= 010
3		RD <= ALUOut	ALUSelA <= 1 ALUSelB <= 100 RegDst <= 01 MemToReg <= 00 RegWrite_H <= 1

sll	Type **R-Type**	Op (31:26) **000000**	Func (5:0) **000000**

Cycle	**Label**	**RTL**	**Control**
2	SLL:	ALUOut <= RT << IR[10-6]	ALUSelA <= 1 ALUSelB <= 100
3		RD <= ALUOut	ALUSelA <= 1 ALUSelB <= 100 RegDst <= 01 MemToReg <= 00 RegWrite_H <= 1

slt

Type	Op (31:26)	Func (5:0)
R-Type	**000000**	**101010**

Cycle	**Label**	**RTL**	**Control**
2	SLT:	if (RS < RT) then ALUOut = 1 else ALUOut = 0	ALUSelA <= 1 ALUSelB <= 000
3		RD <= ALUOut	ALUSelA <= 1 ALUSelB <= 000 RegDst <= 01 MemToReg <= 00 RegWrite_H <= 1

slti

Type	Op (31:26)
I-Type	**101010**

Cycle	**Label**	**RTL**	**Control**
2	SLTI:	if (RS < SEx(IMM)) then ALUOut = 1 else ALUOut = 0	ALUSelA <= 1 ALUSelB <= 010
3		RD <= ALUOut	ALUSelA <= 1 ALUSelB <= 010 RegDst <= 01 MemToReg <= 00 RegWrite_H <= 1

srl

Type	Op (31:26)	Func (5:0)
R-Type	**000000**	**000010**

Cycle	**Label**	**RTL**	**Control**
2	SRL:	ALUOut <= RT >> IR[10-6]	ALUSelA <= 1 ALUSelB <= 100
3		RD <= ALUOut	ALUSelA <= 1 ALUSelB <= 100 RegDst <= 01 MemToReg <= 00 RegWrite_H <= 1

sub

Type	Op (31:26)	Func (5:0)
R-Type	**000000**	**100010**

Cycle	**Label**	**RTL**	**Control**
2	SUB:	ALUOut <= RS - RT	ALUSelA <= 1 ALUSelB <= 000
3		RD <= ALUOut	ALUSelA <= 1 ALUSelB <= 000 RegDst <= 01 MemToReg <= 00 RegWrite_H <= 1

SW

Type	Op (31:26)
I-Type	**100011**

Cycle	**Label**	**RTL**	**Control**
2	SW:	ALUOut <= RS + IMM	ALUSelA <= 1 ALUSelB <= 010
3		Mem[ALUOut] <= RT	MemWrite IorD = 1 ALUSelA <= 1 ALUSelB <= 010 RegDst <= 00 MemToReg <= 01