**COMP4300**
**SimpleRisc Semantics**

**Notes**

1. Memory can perform one read or write per clock cycle
2. Memory can hold 1024 32-bit words
3. Only one ALU operation can be performed per clock cycle
4. A register can store one new value per clock cycle
5. The register file can perform one read or one write per clock cycle
6. The register file holds 32 32-bit numbers

**States**

The machine has eighteen states, numbered 1-18. The state diagram is given by the semantics of each state, in which the next state or states are given

**State 1, all instructions**

Memory[PC]-> IR (instruction register)

**State 2, all instructions**

IR[6-10]-> operand1
IR[11-15]-> operand2
IR[16-20]-> destination
IR[21-24]-> func

If IR[opcode] = 000010 then go to state 3
If IR[opcode] = 000000 then go to state 6
If IR[opcode] = 000001 then go to state 10
If IR[opcode] = 000011 then go to state 14
If IR[opcode] = 000100 then go to state 14
If IR[opcode] = 000101 then go to state 14

**State 3 (start of ALU sequence)**

Regfile[operand1]-> op1r
Go to state 4

**State 4**

Regfile[operand2]-> op2r
Go to state 5

**State 5**

ALU(op1r,op2r,func)->alu_out_reg->Regfile[destination]
PC+1 -> PC
Go to state 1

**State 6 (start of load sequence)**

PC + 1 -> PC
Go to state 7

**State 7**
Memory[PC ]->op2r
Regfile[operand1]-> op1r
Go to state 8

**State 8**
ALU(op1r,op2r,unsigned add)->alu_out_reg
Go to state 9

**State 9**
Memory[alu_out_reg]-> Regfile[destination]
PC+1 -> PC
Go to state 1

**State 10 (start of store sequence)**
PC+1-> PC
Go to state 11

**State 11**
Mem[PC]->op2r
Regfile[destination]->op1r
Go to state 12

**State 12**
ALU(op1r,op2r,unsigned add)-> alu_out_reg
Go to state 13

**State 13**
Regfile[operand1] -> Memory[alu_out_reg]
PC+1->PC
Go to state 1

**State 14 (start of JMP,JZ,JNZ sequence)**
PC+1->PC
Go to state 15

**State 15**
Mem[PC]-> op2r, Regs[operand1] -> op1r
Go to state 16

**State 16**
ALU(op1r,op2r,unsigned add) -> alu_out_reg
Go to state 17

**State 17**
Regfile[operand2]-> Control
Go to state 18

**State 18**
If opcode == 000100 (JZ) then
    If Regfile[operand2] == 0 then
        alu_out_reg -> PC
    else
        PC +1-> PC
Else if opcode == 000101 (JNZ) then
    If Regfile[operand2] != 0 then
        alu_out_reg -> PC
    else
        PC+1-> PC
Else (JMP)
    alu_out_reg -> PC

Go to state 1