**RTL & Control Word**

RTL for RESET : ;

RESET: begin

W\_Adr = R\_Adr = S\_Adr = ;

adr\_sel = s\_sel = ir\_ld = ;

pc\_ld = pc\_inc = pc\_sel = ;

mw\_en = rw\_en = alu\_op = ;

{ns\_N, ns\_Z, ns\_C } = ;

status = ;

nextstate = ;

end

RTL for FETCH : ;

FETCH: begin

W\_Adr = R\_Adr = S\_Adr = ;

adr\_sel = s\_sel = ir\_ld = ;

pc\_ld = pc\_inc = pc\_sel = ;

mw\_en = rw\_en = alu\_op = ;

{ns\_N, ns\_Z, ns\_C } = ;

status = ;

nextstate = ;

end

RTL for DECODE: ;

DECODE: begin

W\_Adr = R\_Adr = S\_Adr = ;

adr\_sel = s\_sel = ir\_ld = ;

pc\_ld = pc\_inc = pc\_sel = ;

mw\_en = rw\_en = alu\_op = ;

{ns\_N, ns\_Z, ns\_C } = ;

status = ;

nextstate = ;

end

RTL for ADD : ;

ADD: begin

W\_Adr = R\_Adr = S\_Adr = ;

adr\_sel = s\_sel = ir\_ld = ;

pc\_ld = pc\_inc = pc\_sel = ;

mw\_en = rw\_en = alu\_op = ;

{ns\_N, ns\_Z, ns\_C } = ;

status = ;

nextstate = ;

end

RTL for SUB : ;

SUB: begin

W\_Adr = R\_Adr = S\_Adr = ;

adr\_sel = s\_sel = ir\_ld = ;

pc\_ld = pc\_inc = pc\_sel = ;

mw\_en = rw\_en = alu\_op = ;

{ns\_N, ns\_Z, ns\_C } = ;

status = ;

nextstate = ;

end

RTL for CMP : ;

CMP: begin

W\_Adr = R\_Adr = S\_Adr = ;

adr\_sel = s\_sel = ir\_ld = ;

pc\_ld = pc\_inc = pc\_sel = ;

mw\_en = rw\_en = alu\_op = ;

{ns\_N, ns\_Z, ns\_C } = ;

status = ;

nextstate = ;

end

RTL for MOV : ;

MOV: begin

W\_Adr = R\_Adr = S\_Adr = ;

adr\_sel = s\_sel = ir\_ld = ;

pc\_ld = pc\_inc = pc\_sel = ;

mw\_en = rw\_en = alu\_op = ;

{ns\_N, ns\_Z, ns\_C } = ;

status = ;

nextstate = ;

end

RTL for SHL : ;

SHL: begin

W\_Adr = R\_Adr = S\_Adr = ;

adr\_sel = s\_sel = ir\_ld = ;

pc\_ld = pc\_inc = pc\_sel = ;

mw\_en = rw\_en = alu\_op = ;

{ns\_N, ns\_Z, ns\_C } = ;

status = ;

nextstate = ;

end

RTL for SHR : ;

SHR: begin

W\_Adr = R\_Adr = S\_Adr = ;

adr\_sel = s\_sel = ir\_ld = ;

pc\_ld = pc\_inc = pc\_sel = ;

mw\_en = rw\_en = alu\_op = ;

{ns\_N, ns\_Z, ns\_C } = ;

status = ;

nextstate = ;

end

RTL for INC : ;

INC: begin

W\_Adr = R\_Adr = S\_Adr = ;

adr\_sel = s\_sel = ir\_ld = ;

pc\_ld = pc\_inc = pc\_sel = ;

mw\_en = rw\_en = alu\_op = ;

{ns\_N, ns\_Z, ns\_C } = ;

status = ;

nextstate = ;

end

RTL for DEC : ;

DEC: begin

W\_Adr = R\_Adr = S\_Adr = ;

adr\_sel = s\_sel = ir\_ld = ;

pc\_ld = pc\_inc = pc\_sel = ;

mw\_en = rw\_en = alu\_op = ;

{ns\_N, ns\_Z, ns\_C } = ;

status = ;

nextstate = ;

end

RTL for LD : ;

LD: begin

W\_Adr = R\_Adr = S\_Adr = ;

adr\_sel = s\_sel = ir\_ld = ;

pc\_ld = pc\_inc = pc\_sel = ;

mw\_en = rw\_en = alu\_op = ;

{ns\_N, ns\_Z, ns\_C } = ;

status = ;

nextstate = ;

end

RTL for STO : ;

STO: begin

W\_Adr = R\_Adr = S\_Adr = ;

adr\_sel = s\_sel = ir\_ld = ;

pc\_ld = pc\_inc = pc\_sel = ;

mw\_en = rw\_en = alu\_op = ;

{ns\_N, ns\_Z, ns\_C } = ;

status = ;

nextstate = ;

end

RTL for LDI : ;

LDI: begin

W\_Adr = R\_Adr = S\_Adr = ;

adr\_sel = s\_sel = ir\_ld = ;

pc\_ld = pc\_inc = pc\_sel = ;

mw\_en = rw\_en = alu\_op = ;

{ns\_N, ns\_Z, ns\_C } = ;

status = ;

nextstate = ;

end

RTL for JE : ;

JE: begin

W\_Adr = R\_Adr = S\_Adr = ;

adr\_sel = s\_sel = ir\_ld = ;

pc\_ld = pc\_inc = pc\_sel = ;

mw\_en = rw\_en = alu\_op = ;

{ns\_N, ns\_Z, ns\_C } = ;

status = ;

nextstate = ;

end

RTL for JNE : ;

JNE: begin

W\_Adr = R\_Adr = S\_Adr = ;

adr\_sel = s\_sel = ir\_ld = ;

pc\_ld = pc\_inc = pc\_sel = ;

mw\_en = rw\_en = alu\_op = ;

{ns\_N, ns\_Z, ns\_C } = ;

status = ;

nextstate = ;

end

RTL for JC : ;

JC: begin

W\_Adr = R\_Adr = S\_Adr = ;

adr\_sel = s\_sel = ir\_ld = ;

pc\_ld = pc\_inc = pc\_sel = ;

mw\_en = rw\_en = alu\_op = ;

{ns\_N, ns\_Z, ns\_C } = ;

status = ;

nextstate = ;

end

RTL for JMP : ;

JMP: begin

W\_Adr = R\_Adr = S\_Adr = ;

adr\_sel = s\_sel = ir\_ld = ;

pc\_ld = pc\_inc = pc\_sel = ;

mw\_en = rw\_en = alu\_op = ;

{ns\_N, ns\_Z, ns\_C } = ;

status = ;

nextstate = ;

end

RTL for HALT : ;

HALT: begin

W\_Adr = R\_Adr = S\_Adr = ;

adr\_sel = s\_sel = ir\_ld = ;

pc\_ld = pc\_inc = pc\_sel = ;

mw\_en = rw\_en = alu\_op = ;

{ns\_N, ns\_Z, ns\_C } = ;

status = ;

nextstate = ;

end

RTL for ILLEGAL\_OP : ;

ILLEGAL\_OP: begin

W\_Adr = R\_Adr = S\_Adr = ;

adr\_sel = s\_sel = ir\_ld = ;

pc\_ld = pc\_inc = pc\_sel = ;

mw\_en = rw\_en = alu\_op = ;

{ns\_N, ns\_Z, ns\_C } = ;

status = ;

nextstate = ;

end