Pipelined ReCoP Instructions

T1: if instr\_prev(31..30) == “01” or “10” IR(15..0) <- PM[PC], OP(15..0) <- PM[PC+1], PC <- PC + 2 else IR(15..0) <- operand\_prev, OP(15..0) <- PM[PC], PC <- PC + 1

|  |  |  |
| --- | --- | --- |
| AND Rz Rx Operand | Immediate | T2: OP1 <- Rx, OP2 <- OP(15..0) |
| T3: RZ <- ALU\_OP1 AND ALU\_OP2 |
| AND Rz Rz Rx | Register | T2: OP1 <- Rz, OP2 <- Rx |
| T3: RZ <- ALU\_OP1 AND ALU\_OP2 |
| OR Rz Rx Operand | Immediate | T2: OP1 <- Rx, OP2 <- OP(15..0) |
| T3: RZ <- ALU\_OP1 OR ALU\_OP2 |
| OR Rz Rz Rx | Register | T2: OP1 <- Rx, OP2 <- Rz |
| T3: Rz <- OP1 OR ALU\_OP2 |
| ADD Rz Rx Operand | Immediate | T2: OP1 <- Rx, OP2 <- OP(15..0) |
| T3: Rz <- ALU\_OP1 + ALU\_OP2 |
| ADD Rz Rz Rx | Register | T2: OP1 <- Rx, OP2 <- Rz |
| T3: Rz <- OP1 + OP2 |
| SUBV Rz Rx Operand | Immediate | T2: OP1 <- OP(15..0), OP2 <- Rx |
| T3: Rz <- OP2 - OP1 |
| SUB Rz Operand | Immediate | T2: OP1 <- OP(15..0), OP2 <- Rz |
| T3: OP2 - OP1 |
| LDR Rz #Operand | Immediate | T2: OP1 <- OP(15..0) |
| T3: Rz <- OP1 |
| LDR Rz Rx | Register | T2: OP2 <- Rx |
| T3: Rz <- DM[OP2] |
| LDR Rz $Operand | Direct | T2: OP2 <- OP(15..0) |
| T3: Rz <- DM[OP2] |
| STR Rz #Operand | Immediate | T2: OP1 <- OP(15..0), OP2 <- Rz |
| T3: DM[OP2] <- OP1 |
| STR Rz Rx | Register | T2: OP1 <- Rx, OP2 <- Rz |
| T3: DM[OP2] <- OP1 |
| STR Rx $Operand | Direct | T2: OP1 <- Rx, OP2 <- OP(15..0) |
| T3: DM[OP2] <- OP1 |
| JMP Operand | Immediate | T2: OP1 <- OP(15..0) |
| T3: PC <- OP1 |
| JMP Rx | Register | T2: OP1 <- Rx |
| T3: PC <- OP1 |
| PRESENT Rz Operand | Immediate | T2: OP1 <- OP(15..0), OP2 <- Rz |
| T3: if OP2(15..0) = 0x0000 then PC <- OP1 |
| DATACALL Rz Rx | Register | T2: OP1 <- Rz, OP2 <- Rx |
| T3: DPCR <- OP1 & OP2, DPRR(1) <- '0' |
| DATACALL Rx Operand | Immediate | T2: OP1 <- Rx, OP2 <- OP(15..0) |
| T3: DPCR <- OP1 & OP2, DPRR(1) <- '0' |
| SZ Operand | Immediate | T2: OP1 <- OP(15..0) |
| T3: if Z == '1' then PC <- OP1 |
| MAX Rz Operand | Immediate | T2: OP1 <- Rz, OP2 <- OP(15..0) |
| T3: Rz <- MAX{OP1, OP2} |
| STRPC Operand | Direct | T2: OP1 <- PC(15..0), OP2 <- OP(15..0) |
| T3: DM[OP2] <- OP1 |
| CLFZ | Inherent | T2: |
| T3: Z <- '0' |
| CER | Inherent | T2: |
| T3: ER <- '0' |
| CEOT | Inherent | T2: |
| T3: EOT <- '0' |
| SEOT | Inherent | T2: |
| T3: EOT <- '1' |
| LER Rz | Register | T2: |
| T3: Rz <- ER |
| SSVOP Rx | Register | T2: OP1 <- Rx |
| T3: SVOP <- Rx |
| LSIP Rz | Register | T2: |
| T3: Rz <- SIP |
| SSOP Rx | Register | T2: OP1 <- Rx |
| T3: SOP <- OP1 |
| NOOP | Inherent | T2: |
| T3: |