# Registers

* DI (32bit)
* RI (32bit)
* BI (32bit)
* S (32bit)
* C1 (32bit)
* C2 (32bit)
* PC (32bit)
* AR (32bit)
* IR (32bit)

# Instruction Set Architecture

|  |  |  |
| --- | --- | --- |
| Opcode | Task | decimal |
| LDAC Γ | AC <= Γ |  |
| CLAC | AC <= 0, Z <= 1 |  |
| INCAC | AC <= AC + 1; if(AC+1=0) Z <= 1 else Z <= 0 |  |
| DECAC | AC <= AC - 1; if(AC+1=0) Z <= 1 else Z <= 0 |  |
| MVAC\_DI | DI <= AC |  |
| MVAC\_RI | RI <= AC |  |
| MVAC\_BI | BI <= AC |  |
| MVAC\_S | S <= AC |  |
| MVAC\_C1 | C1 <= AC |  |
| MVAC\_C2 | C2 <= AC |  |
| MVAC\_AR | AR <= AC |  |
| MVDI | AC <= DI |  |
| MVRI | AC <= RI |  |
| MVBI | AC <= BI |  |
| MVS | AC <= S |  |
| MVC1 | AC <= C1 |  |
| MVC2 | AC <= C2 |  |
| MVAR | AC <= AR |  |
| READ | AC <= M[AR] |  |
| WRITE | M[AR] <= AC |  |
| ADD\_DI | AC <= AC + DI; if(AC+DI=0) Z <= 1 else Z <= 0 |  |
| ADD\_RI | AC <= AC + RI; if(AC+RI=0) Z <= 1 else Z <= 0 |  |
| ADD\_BI | AC <= AC + BI; if(AC+BI=0) Z <= 1 else Z <= 0 |  |
| ADD\_S | AC <= AC + S; if(AC+S=0) Z <= 1 else Z <= 0 |  |
| ADD\_C1 | AC <= AC + C1; if(AC+C1=0) Z <= 1 else Z <= 0 |  |
| ADD\_C2 | AC <= AC + C1; if(AC+C2=0) Z <= 1 else Z <= 0 |  |
| ADD\_AR | AC <= AC + AR; if(AC+AR=0) Z <= 1 else Z <= 0 |  |
| SUB\_DI | AC <= AC - DI; if(AC-DI=0) Z <= 1 else Z <= 0 |  |
| SUB\_RI | AC <= AC - RI; if(AC-RI=0) Z <= 1 else Z <= 0 |  |
| SUB\_BI | AC <= AC - BI; if(AC-BI=0) Z <= 1 else Z <= 0 |  |
| SUB\_S | AC <= AC - S; if(AC-S=0) Z <= 1 else Z <= 0 |  |
| SUB\_C1 | AC <= AC – C1; if(AC-C1=0) Z <= 1 else Z <= 0 |  |
| SUB\_C2 | AC <= AC – C2; if(AC-C2=0) Z <= 1 else Z <= 0 |  |
| SUB\_AR | AC <= AC - AR; if(AC-AR=0) Z <= 1 else Z <= 0 |  |
| MUL2 | AC <= AC << 1; if(AC << 1=0) Z <= 1 else Z <= 0 |  |
| MUL4 | AC <= AC << 2; if(AC << 2=0) Z <= 1 else Z <= 0 |  |
| DIV16 | AC <= AC >> 4; if(AC >> 4=0) Z <= 1 else Z <= 0 |  |
| JMPZ Γ | If(z=0) goto Γ |  |
| JMPNZ Γ | If(z!=0) goto Γ |  |
| END | PC <= 0; FINISH <= 1 |  |
| NOP | Do nothing |  |