

| Mnemonic                   | Semantics                                  | Encoding           |
|----------------------------|--|--------------------|
| <b>Instructions</b>        |  |                    |
| NOP                        | --   | 0000000-----       |
| BRn IMM9 <LABEL>           | N ? PC = PC+1+SEXT (IMM9)                  | 0000100IIIIIIIIII  |
| BRnz IMM9 <LABEL>          | N Z ? PC = PC+1+SEXT (IMM9)                | 0000110IIIIIIIIII  |
| BRnp IMM9 <LABEL>          | N P ? PC = PC+1+SEXT (IMM9)                | 0000101IIIIIIIIII  |
| BRz IMM9 <LABEL>           | Z ? PC = PC+1+SEXT (IMM9)                  | 0000010IIIIIIIIII  |
| BRzp IMM9 <LABEL>          | Z P ? PC = PC+1+SEXT (IMM9)                | 0000011IIIIIIIIII  |
| BRp IMM9 <LABEL>           | P ? PC = PC+1+SEXT (IMM9)                  | 0000001IIIIIIIIII  |
| BRnzp IMM9 <LABEL>         | PC = PC+1+SEXT (IMM9)                      | 0000111IIIIIIIIII  |
| ADD Rd, Rs, Rt             | Rd = Rs+Rt                                 | 0001dddsss000ttt   |
| MUL Rd, Rs, Rt             | Rd = Rs*Rt                                 | 0001dddsss001ttt   |
| SUB Rd, Rs, Rt             | Rd = Rs-Rt                                 | 0001dddsss010ttt   |
| DIV Rd, Rs, Rt             | Rd = Rs/Rt                                 | 0001dddsss011ttt   |
| ADD Rd, Rs, IMM5           | Rd = Rs+SEXT (IMM5)                        | 0001dddsss1IIIII   |
| CMP Rs, Rt                 | NZP = signed-CC (Rs-Rt)                    | 0010sss00----ttt   |
| CMPU Rs, Rt                | NZP = unsigned-CC (Rs-Rt)                  | 0010sss01----ttt   |
| CMPI Rs, IMM7              | NZP = signed-CC (Rs-SEXT (IMM7))           | 0010sss10IIIIIIII  |
| CMPIU Rs, UIMM7            | NZP = unsigned-CC (Rs-UIMM7)               | 0010sss11UUUUUUU   |
| JSR IMM11 <LABEL>          | R7 = PC+1; PC = (PC&x8000)   (IMM11<<4)    | 01001IIIIIIIIIIII  |
| JSRR Rs                    | R7 = PC+1; PC = Rs                         | 01000--sss-----    |
| AND Rd, Rs, Rt             | Rd = Rs&Rt                                 | 0101dddsss000ttt   |
| NOT R1, R2                 | Rd = Rs                                    | 0101dddsss001---   |
| OR R1, R2, R3              | Rd = Rs Rt                                 | 0101dddsss010ttt   |
| XOR R1, R2, R3             | Rd = Rs^Rt                                 | 0101dddsss011ttt   |
| AND Rd, Rs, IMM5           | Rd = Rs&SEXT (IMM5)                        | 0101dddsss1IIIII   |
| LDR Rd, Rs, IMM6           | Rd = dmem [Rs+SEXT (IMM6) ]                | 0110dddsssIIIIII   |
| STR Rd, Rs, IMM6           | dmem [Rs+SEXT (IMM6) ] = Rd                | 0111dddsssIIIIII   |
| RTI                        | PC = R7; PSR[15] = 0                       | 1000-----          |
| CONST Rd, IMM9             | Rd = SEXT (IMM9)                           | 1001dddIIIIIIIIII  |
| SLL Rd, Rs, UIMM4          | Rd = Rs<<UIMM4                             | 1010dddsss00UUUU   |
| SRA Rd, Rs, UIMM4          | Rd = Rs>>>UIMM4                            | 1010dddsss01UUUU   |
| SRL Rd, Rs, UIMM4          | Rd = Rs>>UIMM4                             | 1010dddsss10UUUU   |
| MOD Rd, Rs, Rt             | Rd = Rs%Rt                                 | 1010dddsss11-ttt   |
| JMPR Rs                    | PC = Rs                                    | 11000--sss-----    |
| JMP IMM11 <LABEL>          | PC = PC+1+SEXT (IMM11)                     | 11001iiiiiiiiiiii  |
| HICONST Rd, UIMM8          | Rd = (Rd&xFF)   (UIMM8<<8)                 | 1101ddd1UUUUUUUU   |
| TRAP UIMM8                 | R7 = PC+1; PC = (x8000 UIMM8); PSR[15] = 1 | 1111----UUUUUUUU   |
| <b>Pseudo-instructions</b> |  |                    |
| RET                        | JMPR R7                                    |                    |
| LEA R1, <LABEL>            | R1 = address of label                      |                    |
| LC R1, <LABEL>             | R1 = constant at label                     |                    |
| <b>Assembly directives</b> |  |                    |
| .DATA                      | current memory is data                     |                    |
| .CODE                      | current memory is code                     |                    |
| .ADDR UIMM16               | set current address to UIMM16              |                    |
| .FALIGN                    | pad current address to 16-word boundary    |                    |
| .FILL IMM16                | set value at current address to IMM16      | IIIIIIIIIIIIIIIIII |
| .BLKW UIMM16               | reserve UIMM16 words at current address    | 0000000000000000   |
| .CONST IMM16               | associate IMM16 with preceding label       |                    |
| .UNCONST UIMM16            | associate UIMM16 with preceding label      |                    |

001: opcode or sub-opcode

ddd: d-register, sss: s-register, ttt: t-register

III: signed immediate, UUU: unsigned immediate, ---: don't care