

```
-- File Mopcodes.Mesa
-- Last modified by Johnsson; May 18, 9177 11:33 AM
```

```
Mopcodes:      DEFINITIONS =
BEGIN
```

```
opcode:        TYPE = [0..377B];
```

```
-- LOAD/STORE (16 bits) [10B..67B]
```

```
zLGn,zLG0:    opcode = 10B;
zLG1:         opcode = zLG0 + 1;
zLG2:         opcode = zLG1 + 1;
zLG3:         opcode = zLG2 + 1;
zLG4:         opcode = zLG3 + 1;
zLG5:         opcode = zLG4 + 1;
zLG6:         opcode = zLG5 + 1;
zLG7:         opcode = zLG6 + 1;
zLGB:         opcode = zLG7 + 1;
zLGS:         opcode = zLGB + 1;
zSGn,zSG0:    opcode = zLGS + 1;
zSG1:         opcode = zSG0 + 1;
zSG2:         opcode = zSG1 + 1;
zSG3:         opcode = zSG2 + 1;
zSGB:         opcode = zSG3 + 1;
zSGS:         opcode = zSGB + 1;
zLLn,zLL0:    opcode = zSGS + 1;
zLL1:         opcode = zLL0 + 1;
zLL2:         opcode = zLL1 + 1;
zLL3:         opcode = zLL2 + 1;
zLL4:         opcode = zLL3 + 1;
zLL5:         opcode = zLL4 + 1;
zLL6:         opcode = zLL5 + 1;
zLL7:         opcode = zLL6 + 1;
zLLB:         opcode = zLL7 + 1;
zLLS:         opcode = zLLB + 1;
zSLn,zSL0:    opcode = zLLS + 1;
zSL1:         opcode = zSL0 + 1;
zSL2:         opcode = zSL1 + 1;
zSL3:         opcode = zSL2 + 1;
zSL4:         opcode = zSL3 + 1;
zSL5:         opcode = zSL4 + 1;
zSL6:         opcode = zSL5 + 1;
zSL7:         opcode = zSL6 + 1;
zSLB:         opcode = zSL7 + 1;
zSLS:         opcode = zSLB + 1;
zLIIn,zLI0:   opcode = zSLS + 1;
zLI1:         opcode = zLI0 + 1;
zLI2:         opcode = zLI1 + 1;
zLI3:         opcode = zLI2 + 1;
zLI4:         opcode = zLI3 + 1;
zLI5:         opcode = zLI4 + 1;
zLI6:         opcode = zLI5 + 1;
zLINn,zLIN1:  opcode = zLI6 + 1;
zLIB:         opcode = zLIN1 + 1;
zLIW:         opcode = zLIB + 1;
zLINB:        opcode = zLIW + 1;
```

```
-- LOAD/STORE (32 bits) [70B..77B]
```

```
zLGDB:        opcode = 70B;
zLGDS:        opcode = zLGDB + 1;
zSGDB:        opcode = zLGDS + 1;
zSGDS:        opcode = zSGDB + 1;
zLIDB:        opcode = zSGDS + 1;
zLIDS:        opcode = zLIDB + 1;
zSIDB:        opcode = zLIDS + 1;
zSIDS:        opcode = zSIDB + 1;
```

```
-- READ/WRITE using TOS as pointer [100B..127B]
```

```
zRn,zR0:     opcode = 100B;
zR1:         opcode = zR0 + 1;
zR2:         opcode = zR1 + 1;
zR3:         opcode = zR2 + 1;
```

```

zR4:          opcode = zR3 + 1;
zRB:          opcode = zR4 + 1;
zWn, zW0:     opcode = zRB + 1;
zW1:          opcode = zW0 + 1;
zW2:          opcode = zW1 + 1;
zWB:          opcode = zW2 + 1;
zRF:          opcode = zWB + 1;
zWF:          opcode = zRF + 1;
zRDB:         opcode = zWF + 1;
zRDO:         opcode = zRDB + 1;
zWDB:         opcode = zRDO + 1;
zWDO:         opcode = zWDB + 1;
zRSTR:        opcode = zWDO + 1;
zWSTR:        opcode = zRSTR + 1;

```

-- READ/WRITE using local as tied-pointer base [122B..126B]

```

zRXLn, zRXLO: opcode = 122B;
zRXL1:        opcode = zRXLO + 1;
zRXL2:        opcode = zRXL1 + 1;
zRXL3:        opcode = zRXL2 + 1;
zWXLn, zWXL0: opcode = zRXL3 + 1;

```

-- READ/WRITE using local/global as pointer [127B..147B]

```

zRIGn, zRIG0: opcode = 127B;
zRIG1:        opcode = zRIG0 + 1;
zRIG2:        opcode = zRIG1 + 1;
zRIG3:        opcode = zRIG2 + 1;
zWIGn, zWIG0: opcode = zRIG3 + 1;
zRILn, zRILO: opcode = zWIG0 + 1;
zRIL1:        opcode = zRILO + 1;
zRIL2:        opcode = zRIL1 + 1;
zRIL3:        opcode = zRIL2 + 1;
zWILn, zWILO: opcode = zRIL3 + 1;
zWS0:         opcode = zWILO + 1;
zWSB:         opcode = zWS0 + 1;
zWSF:         opcode = zWSB + 1;
zWSDB:        opcode = zWSF + 1;
zWSDS:        opcode = zWSDB + 1;

```

-- JUMPS [150B..247B]

```

zJnE, zJ1E:   opcode = 150B;
zJ2E:         opcode = zJ1E + 1;
zJ3E:         opcode = zJ2E + 1;
zJ4E:         opcode = zJ3E + 1;
zJn0, zJ10:   opcode = zJ4E + 1;
zJ20:         opcode = zJ10 + 1;
zJ30:         opcode = zJ20 + 1;
zJ40:         opcode = zJ30 + 1;
zJBE:         opcode = zJ40 + 1;
zJBO:         opcode = zJBE + 1;
zJWE:         opcode = zJBO + 1;
zJWO:         opcode = zJWE + 1;
zNJBE:        opcode = zJWO + 1;
zNJBO:        opcode = zNJBE + 1;
zJEqnE, zJEq1E: opcode = zNJBO + 1;
zJEq2E:       opcode = zJEq1E + 1;
zJEq3E:       opcode = zJEq2E + 1;
zJEq4E:       opcode = zJEq3E + 1;
zJEqn0, zJFQ10: opcode = zJEq4E + 1;
zJFQ20:       opcode = zJFQ10 + 1;
zJFQ30:       opcode = zJFQ20 + 1;
zJFQ40:       opcode = zJFQ30 + 1;
zJFQ8F:       opcode = zJFQ40 + 1;
zJFQ80:       opcode = zJFQ8F + 1;
zJNFnF, zJNF1F: opcode = zJFQ80 + 1;
zJNF2F:       opcode = zJNF1F + 1;
zJNF3F:       opcode = zJNF2F + 1;
zJNF4F:       opcode = zJNF3F + 1;
zJNFn0, zJNF10: opcode = zJNF4F + 1;
zJNF20:       opcode = zJNF10 + 1;
zJNF30:       opcode = zJNF20 + 1;
zJNF40:       opcode = zJNF30 + 1;
zJNF8F:       opcode = zJNF40 + 1;

```

```

zJNEBO:      opcode = zJNEBE + 1;
zJLBE:      opcode = zJNEBO + 1;
zJLBO:      opcode = zJLBE + 1;
zJGEBE:     opcode = zJLBO + 1;
zJGEBO:     opcode = zJGEBE + 1;
zJGBE:      opcode = zJGEBO + 1;
zJGBO:      opcode = zJGBE + 1;
zJLEBE:     opcode = zJGBO + 1;
zJLEBO:     opcode = zJLEBE + 1;
zJULBE:     opcode = zJLEBO + 1;
zJULBO:     opcode = zJULBE + 1;
zJUGEBE:    opcode = zJULBO + 1;
zJUGEBO:    opcode = zJUGEBE + 1;
zJUGBE:     opcode = zJUGEBO + 1;
zJUGBO:     opcode = zJUGBE + 1;
zJULEBE:    opcode = zJUGBO + 1;
zJULEBO:    opcode = zJULEBE + 1;
zJZQEBO:   opcode = zJULEBO + 1;
zJZQEBO:   opcode = zJZQEBO + 1;
zJZNEBE:   opcode = zJZQEBO + 1;
zJZNEBO:   opcode = zJZNEBE + 1;
zJDEQBE:   opcode = zJZNEBO + 1;
zJDEQBO:   opcode = zJDEQBE + 1;
zJDNEBE:   opcode = zJDEQBO + 1;
zJDNEBO:   opcode = zJDNEBE + 1;
zJIB:      opcode = zJDNEBO + 1;
zJIW:      opcode = zJIB + 1;

```

-- ARITHMETIC and LOGICAL OPERATORS [250B..267B]

```

zADD:      opcode = 250B;
zSUB:      opcode = zADD + 1;
zMUL:      opcode = zSUB + 1;
zDBL:      opcode = zMUL + 1;
zDIV:      opcode = zDBL + 1;
zLDIV:     opcode = zDIV + 1;
zNEG:      opcode = zLDIV + 1;
zINC:      opcode = zNEG + 1;
zAND:      opcode = zINC + 1;
zOR:       opcode = zAND + 1;
zXOR:      opcode = zOR + 1;
zSHIFT:    opcode = zXOR + 1;

```

-- STACK maintenance and R-register access [270B..277B]

```

zPUSH:     opcode = 270B;
zPOP:      opcode = zPUSH + 1;
zEXCH:     opcode = zPOP + 1;
zPUSHX:    opcode = zEXCH + 1;
zDUP:      opcode = zPUSHX + 1;

```

-- PROCEDURE CALLS [300B..347B]

```

zGFCn,zGFC0: opcode = 300B;
zGFC1:      opcode = zGFC0 + 1;
zGFC2:      opcode = zGFC1 + 1;
zGFC3:      opcode = zGFC2 + 1;
zGFC4:      opcode = zGFC3 + 1;
zGFC5:      opcode = zGFC4 + 1;
zGFC6:      opcode = zGFC5 + 1;
zGFC7:      opcode = zGFC6 + 1;
zGFC8:      opcode = zGFC7 + 1;
zGFC9:      opcode = zGFC8 + 1;
zGFC10:     opcode = zGFC9 + 1;
zGFC11:     opcode = zGFC10 + 1;
zGFC12:     opcode = zGFC11 + 1;
zGFC13:     opcode = zGFC12 + 1;
zGFC14:     opcode = zGFC13 + 1;
zGFC15:     opcode = zGFC14 + 1;
zGFCB:      opcode = zGFC15 + 1;
zLFCn,zLFC1: opcode = zGFCB + 1;
zLFC2:      opcode = zLFC1 + 1;
zLFC3:      opcode = zLFC2 + 1;
zLFC4:      opcode = zLFC3 + 1;
zLFC5:      opcode = zLFC4 + 1;
zLFC6:      opcode = zLFC5 + 1;

```

```
zLFC7:      opcode = zLFC6 + 1;
zLFC8:      opcode = zLFC7 + 1;
zLFC9:      opcode = zLFC8 + 1;
zLFC10:     opcode = zLFC9 + 1;
zLFC11:     opcode = zLFC10 + 1;
zLFC12:     opcode = zLFC11 + 1;
zLFC13:     opcode = zLFC12 + 1;
zLFC14:     opcode = zLFC13 + 1;
zLFC15:     opcode = zLFC14 + 1;
zLFC16:     opcode = zLFC15 + 1;
zLFCB:      opcode = zLFC16 + 1;
zSFC:       opcode = zLFCB + 1;
zRET:       opcode = zSFC + 1;
zXFER:      opcode = zRET + 1;
zPORTO:     opcode = zXFER + 1;
zPORTI:     opcode = zPORTO + 1;
zKFCB:      opcode = zPORTI + 1;
```

-- MISCELLANEOUS [350B..357B]

```
zADDL:      opcode = 350B;
zADDG:      opcode = zADDL + 1;
zBLT:       opcode = zADDG + 1;
zALLOC:     opcode = zBLT + 1;
zFREE:      opcode = zALLOC + 1;
zIWDC:      opcode = zFREE + 1;
zDWDC:      opcode = zIWDC + 1;
```

-- OS MACHINERY [360B..367B]

```
zSTOP:      opcode = 360B;
zCATCH:     opcode = zSTOP + 1;
zCONVERT:   opcode = zCATCH+1;
zBLOCK:     opcode = zCONVERT+1;
zBITBLT:    opcode = zBLOCK+1;
zSTARTIO:   opcode = zBITBLT +1;
```

-- STATE MANIPULATION [370B..377B]

```
zDST:       opcode = 370B;
zLST:       opcode = 371B;
zLSTF:      opcode = 372B;
zWR:        opcode = 374B;
zRR:        opcode = 375B;
zBRK:       opcode = 376B;
```

END...