## 3.6.1 REGISTER TO REGISTER

| Mnemonic | Operand | Cycles | Instruction | | | | | | Description | Status Change |
|---|---|---|---|---|---|---|---|---|---|---|
| MOVR | SSS,DDD | 6/7* | X | XXX | XX0 | 010 | SSS | DDD | MOVe contents of Register SSS to Register DDD. | S, Z |
| TSTR | SSS | 6/7** | | | 0 | 010 | SSS | SSS | TeST contents of Register SSS. | S, Z |
| JR | SSS | 7 | | | 0 | 010 | SSS | 111 | Jump to address in Register SSS (move address to Register 7). | S, Z |
| ADDR | SSS,DDD | 6/7* | | | 0 | 011 | SSS | DDD | ADD contents of Register SSS to contents of register DDD. Results to DDD. | S, Z, C, OV |
| SUBR | SSS,DDD | 6/7* | | | 0 | 100 | SSS | DDD | SUBtract contents of Register SSS from contents of register DDD. Results to DDD. | S, Z, C, OV |
| CMPR | SSS,DDD | 6/7* | | | 0 | 101 | SSS | DDD | CoMPare Register SSS with register DDD by subtraction. Results not stored. | S, Z, C, OV |
| ANDR | SSS,DDD | 6/7* | | | 0 | 110 | SSS | DDD | logical AND contents of Register SSS with contents of register DDD. Results to DDD. | S, Z |
| XORR | SSS,DDD | 6/7* | | | 0 | 111 | SSS | DDD | eXclusive OR contents of Register SSS with contents of register DDD. Results to DDD. | S, Z |
| CLRR | DDD | 6/7* | | | 0 | 111 | DDD | DDD | CLeaR Register to zero. | S, Z |

*7 Cycles if DDD=6 or 7.

**7 Cycles if SSS=6 or 7.

## 3.6.2 SINGLE REGISTER

| Mnemonic | Operand | Cycles | Instruction | | | | | | Description | Status Change |
|---|---|---|---|---|---|---|---|---|---|---|
| INCR | DDD | 6/7* | X | XXX | XX0 | 000 | 001 | DDD | INCrement contents of Register DDD. Results to DDD. | S, Z |
| DECR | DDD | 6/7** | | | 0 | 000 | 010 | DDD | DECrement contents of Register DDD. Results to DDD. | S, Z |
| COMR | DDD | 7 | | | 0 | 000 | 011 | DDD | one's COMplement contents of Register DDD. Results to DDD. | S, Z |
| NEGR | DDD | 6/7* | | | 0 | 000 | 100 | DDD | two's complement contents of Register DDD. Results to DDD. | S, Z, C, OV |
| ADCR | DDD | 6/7* | | | 0 | 000 | 101 | DDD | ADd Carry bit to contents of Register DDD. Results to DDD. | S, Z, C, OV |
| GSWD | DDD | 6 | | | 0 | 000 | 110 | 0DD | Get Status WorD in register DD. Bits 0-3, 8-11 set to 0. Bits 4, 12=C; 5, 13=OV; 6, 14=Z; 7, 15=S. | |
| NOP | | 6 | | | 0 | 000 | 110 | 10X | No Operation. | |
| SIN | | 6 | | | 0 | 000 | 110 | 11X | Software Interrupt; pulse to PCIT pin. | |
| RSWD | SSS | 6 | | | 0 | 000 | 111 | SSS | Restore Status WorD from register SSS; Bit 4 to C, Bit 5 to OV, Bit 6 to Z, Bit 7 to S. | S, Z, C, OV |

## 3.6.3 REGISTER SHIFT

Executable only with Registers 0, 1, 2, 3.
Shifts are not interruptible.

| Mnemonic | Operand | Cycles | Instruction | Description | Status Change |
|---|---|---|---|---|---|
| SWAP | RR<,n> | 6 | X XXX XX0 001 001 NRR | N=0, SWAP bytes of register RR. S equals Bit 7 of results of SWAP. | S, Z |
| | | 8 | | N=1, not supported. | S, Z |
| SLL | RR<,n> | 6 | 0 001 001 NRR | N=0, Shift Logical Left one bit. Zero to low bit. | S, Z |
| | | 8 | | N=1, Shift Logical Left two bits. Zero to low 2 bits. | S, Z |
| RLC | RR<,n> | 6 | 0 001 010 NRR | N=0, Rotate Left one bit using C as bit 16. | S, Z, C |
| | | 8 | | N=1, Rotate Left two bits using C as bit 17 and OV as bit 16. | S, Z, C, OV |
| SLLC | RR<,n> | 6 | 0 001 011 NRR | N=0, Shift Logical Left one bit using C as bit 16. Zero to low bit. | S, Z, C |
| | | 8 | | N=1, Shift Logical Left two bits using C as bit 17, OV as bit 16. Zero to low 2 bits. | S, Z, C, OV |
| SLR | RR<,n> | 6 | 0 001 100 NRR | N=0, Shift Logical Right one bit. Zero to high bit. | S, Z |
| | | 8 | | N=1, Shift Logical Right two bits. Zero to high two bits. | S, Z |
| SAR | RR<,n> | 6 | 0 001 101 NRR | N=0, Shift Arithmetic Right one bit. Sign bit copied to high bit. | S, Z |
| | | 8 | | N=1, Shift Arithmetic Right two bits. Sign bit copied to high 2 bits. | S, Z |
| RRC | RR<,n> | 6 | 0 001 110 NRR | N=0, Rotate right one bit using C as bit 16. | S, Z, C |
| | | 8 | | N=1, Rotate right two bits using C as bit 16 and OV as bit 17. | S, Z, C, OV |
| SARC | RR<,n> | 6 | 0 001 111 NRR | N=0, Shift Arithmetic Right one bit, thru C. Sign bit copied to high bit. | S, Z, C |
| | | 8 | | N=1, Shift Arithmetic Right two bits, thru OV and C. Sign bit copied to high 2 bits. | S, Z, C, OV |

NOTE: n = 1 or 2 places

## 3.6.4 CONTROL

| Mnemonic | Operand | Cycles | Instruction | Description | Status Change |
|---|---|---|---|---|---|
| HLT | | 4 | X XXX XX0 000 000 000 | HaLT after next interruptible instruction is executed. Resume on start | |
| SDBD | | 4 | 0 000 000 001 | Set Double Byte Data for the next instruction which must be an external reference instruction. | |
| EIS | | 4 | 0 000 000 010 | Enable Interrupt System. Not Interruptable. | |
| DIS | | 4 | 0 000 000 011 | Disable Interrupt System. Not Interruptable. | |
| TCI | | 4 | 0 000 000 101 | Terminate Current Interrupt. Not Interruptable. | |
| CLRC | | 4 | 0 000 000 110 | CLeaR Carry to zero. Not Interruptable. | C |
| SETC | | 4 | 0 000 000 111 | SET Carry to one. Not interruptable. | C |

## 3.6.5 JUMP

| Mnemonic | Operand | Cycles | Instruction | Description | Status Change |
|---|---|---|---|---|---|
| J | DA | | X XXX XX0 000 000 100 | Jump to address. Program counter is set to 16 bits of A's. | |
| | | | X XXX XX1 1AA AAA A00 | | |
| | | | X XXX XXA AAA AAA AAA | | |
| JE | DA | | 0 000 000 100 | Jump to address. Enable interrupt system. Program counter is | |
| | | | 1 1AA AAA A01 | set to 16 bits of A's. | |
| | | | A AAA AAA AAA | | |
| JD | DA | | 0 000 000 100 | Jump to address. Disable interrupt system. Program counter is | |
| | | | 1 1AA AAA A10 | set to 16 bits of A's. | |
| | | | A AAA AAA AAA | | |
| JSR | BB, DA | | 1 000 000 100 | Jump and Save Return address (PC + 3) in register designated by | |
| | | | B BAA AAA A00 | 1BB. Program counter is set to 16 bits of A's. BB != 11. | |
| | | | A AAA AAA AAA | | |
| JSRE | BB, DA | | 0 000 000 100 | Jump and Save Return and Enable interrupt system. Return | |
| | | | B BAA AAA A01 | (PC + 3) is saved in register 1BB. Program counter is set to | |
| | | | A AAA AAA AAA | 16 bits of A's. BB != 11. | |
| JSRD | BB, DA | | 0 000 000 100 | Jump and Save Return and Disable interrupt system. Return | |
| | | | B BAA AAA A10 | (PC + 3) is saved in register 1BB. Program counter is set to | |
| | | | A AAA AAA AAA | 16 bits of A's. BB != 11. | |

NOTE: Bits 2-7 of the second word form bits 10-16 of the Destination Address.
Bits 0-9 of the third word form bits 0-9 of the Destination Address.

## 3.6.6 BRANCHES

The Branch instructions are Program Counter Relative, i.e. the Effectrive Address = PC +/- Displacement.
P-P is the Displacement and S is 0 for + and 1 for -.
For a forward branch an addition is performed.
For a backward branch a one's complement subtraction is performed.
Computation performend on PC + 2.

| Mnemonic | Operand | Cycles | Instruction | | | | | | Description | Status Change |
|---|---|---|---|---|---|---|---|---|---|---|
| B | DA | 9 | X | XXX | XX1 | 000 | S00 | 000 | Branch unconditional. Program counter relative (+ 1025 to - 1024) | |
| | | | P | PPP | PPP | PPP | PPP | PPP | | |
| NOPP | DA | 7 | | | 1 | 000 | S01 | 000 | No Operation, two words | |
| | | | P | PPP | PPP | PPP | PPP | PPP | | |
| BC (BLGE) | DA | 7 / 9 | | | 1 | 000 | S00 | 001 | Branch on Carry (Branch if Logical Greater Than or | |
| | | | P | PPP | PPP | PPP | PPP | PPP | Equal). C = 1. | |
| BNC (BLLT) | DA | 7 / 9 | | | 1 | 000 | S01 | 001 | Branch on No Carry (Branch if Logical Less Than). C = 0 | |
| | | | P | PPP | PPP | PPP | PPP | PPP | | |
| BOV | DA | 7 / 9 | | | 1 | 000 | S00 | 010 | Branch on Overflow. OV = 1 | |
| | | | P | PPP | PPP | PPP | PPP | PPP | | |
| BNOV | DA | 7 / 9 | | | 1 | 000 | S01 | 010 | Branch on No Overflow. OV = 0 | |
| | | | P | PPP | PPP | PPP | PPP | PPP | | |
| BPL | DA | 7 / 9 | | | 1 | 000 | S00 | 011 | Branch on Plus. S = 0 | |
| | | | P | PPP | PPP | PPP | PPP | PPP | | |
| BMI | DA | 7 / 9 | | | 1 | 000 | S01 | 011 | Branch on Minus. S = 1 | |
| | | | P | PPP | PPP | PPP | PPP | PPP | | |
| BZE (BEQ) | DA | 7 / 9 | | | 1 | 000 | S00 | 100 | Branch on Zero (Branch if Equal). Z = 1 | |
| | | | P | PPP | PPP | PPP | PPP | PPP | | |
| BNZE (BNEQ) | DA | 7 / 9 | | | 1 | 000 | S01 | 100 | Branch on No Zero (Branch if Not Equal). Z = 0 | |
| | | | P | PPP | PPP | PPP | PPP | PPP | | |
| BLT | DA | 7 / 9 | | | 1 | 000 | S00 | 101 | Branch if Less Than. S ^ OV = 1 | |
| | | | P | PPP | PPP | PPP | PPP | PPP | | |
| BGE | DA | 7 / 9 | | | 1 | 000 | S01 | 101 | Branch if Greater than or Equal. S ^ OV = 0 | |
| | | | P | PPP | PPP | PPP | PPP | PPP | | |
| BLE | DA | 7 / 9 | | | 1 | 000 | S00 | 110 | Branch if Less than or Equal. Z \| (S ^ OV) = 1 | |
| | | | P | PPP | PPP | PPP | PPP | PPP | | |
| BGT | DA | 7 / 9 | | | 1 | 000 | S01 | 110 | Branch if Greater Than. Z \| (S ^ OV) = 0 | |
| | | | P | PPP | PPP | PPP | PPP | PPP | | |
| BUSC | DA | 7 / 9 | | | 1 | 000 | S00 | 111 | Branch if Unequal Sign and Carry. C ^ S = 1 | |
| | | | P | PPP | PPP | PPP | PPP | PPP | | |
| BESC | DA | 7 / 9 | | | 1 | 000 | S01 | 111 | Branch on Equal Sign and Carry. C ^ S = 0 | |
| | | | P | PPP | PPP | PPP | PPP | PPP | | |
| BEXT | DA, EEEE | 7 / 9 | | | 1 | 000 | S1E | EEE | Branch if External condition is True. Field EEEE is externally | |
| | | | P | PPP | PPP | PPP | PPP | PPP | decoded to select 1 of 16 conditions. Response is tested | |
| | | | | | | | | | for true condition. | |

NOTE: 7/9 7 Cycles if test condition is not true, 9 cycles if true.

## 3.6.7 DIRECT ADDRESSED DATA — MEMORY

| Mnemonic | Operand | Cycles | Instruction | | | | | | Description | Status Change |
|---|---|---|---|---|---|---|---|---|---|---|
| MVO | SSS, DA | 11 | X | XXX | XX1 | 001 | 000 | SSS | MoVe Out data from register SSS to address A - A. Not | |
| | | | A | AAA | AAA | AAA | AAA | AAA | interruptible. | |
| MVI | SA, DDD | 10 | | | 1 | 010 | 000 | DDD | MoVe In data from address A - A to register DDD. | |
| | | | A | AAA | AAA | AAA | AAA | AAA | | |
| ADD | SA, DDD | 10 | | | 1 | 011 | 000 | DDD | ADD data from address A - A to register DDD. Results DDD. | S, Z, C, OV |
| | | | A | AAA | AAA | AAA | AAA | AAA | | |
| SUB | SA, DDD | 10 | | | 1 | 100 | 000 | DDD | SUBtract data from address A - A from register DDD. | S, Z, C, OV |
| | | | A | AAA | AAA | AAA | AAA | AAA | Results to DDD. | |
| CMP | SA, DDD | 10 | | | 1 | 101 | 000 | DDD | CoMPare data from address A-A with register SSS by subtraction. | S, Z, C, OV |
| | | | A | AAA | AAA | AAA | AAA | AAA | Results not stored. | |
| AND | SA, DDD | 10 | | | 1 | 110 | 000 | DDD | logical AND data from address A - A with register DDD. | S, Z |
| | | | A | AAA | AAA | AAA | AAA | AAA | Results to DDD. | |
| XOR | SA, DDD | 10 | | | 1 | 111 | 000 | DDD | eXclusive OR data from address A - A with register DDD. | S, Z |
| | | | A | AAA | AAA | AAA | AAA | AAA | Results to DDD. | |

## 3.6.8 IMMEDIATE DATA — REGISTER

| Mnemonic | Operand | Cycles | Instruction | | | | | | Description | Status Change |
|---|---|---|---|---|---|---|---|---|---|---|
| MVOI | SSS, DA | 9 | X | XXX | XX1 | 001 | 000 | SSS | MoVe Out Immediate data from register SSS to PC + 1 (field). | |
| | | | I | III | III | III | III | III | Not interruptible. | |
| MVII | SA, DDD | 8 | | | 1 | 010 | 000 | DDD | MoVe In Immediate data to register DDD from PC + 1 (field). | |
| | | | I | III | III | III | III | III | | |
| ADDI | SA, DDD | 8 | | | 1 | 011 | 000 | DDD | Add Immediate data to contents of register DDD. | S, Z, C, OV |
| | | | I | III | III | III | III | III | Results to DDD. | |
| SUBI | SA, DDD | 8 | | | 1 | 100 | 000 | DDD | SUBtract Immediate data from contents of register DDD. | S, Z, C, OV |
| | | | I | III | III | III | III | III | Results to DDD. | |
| CMPI | SA, DDD | 8 | | | 1 | 101 | 000 | DDD | CoMPare Immediate data from contents of register SSS by | S, Z, C, OV |
| | | | I | III | III | III | III | III | subtraction. Results not stored. | |
| ANDI | SA, DDD | 8 | | | 1 | 110 | 000 | DDD | logical AND Immediate data with contest of register DDD. | S, Z |
| | | | I | III | III | III | III | III | Results to DDD. | |
| XORI | SA, DDD | 8 | | | 1 | 111 | 000 | DDD | eXclusive OR Immediate data with the contents of register DDD. | S, Z |
| | | | I | III | III | III | III | III | Results to DDD. | |

## 3.6.9 INDIRECT ADDRESSED DATA—REGISTE

MMM Source data is located at the address contained in Register R1 - R6.
MMM=4, 5: post-increment R4 or R5.
MMM=6: MVO instruction — post-increment R6. PUSH data from Register SSS to the Stack.
Other instructions — pre-decrement R. PULL data from the Stack to be used at the first operand.

| Mnemonic | Operand | Cycles | Instruction | | | | | | Description | Status Change |
|---|---|---|---|---|---|---|---|---|---|---|
| MVO@ | SSS, MMM | 9 | X | XXX | XX1 | 001 | MMM | SSS | MoVe Out data from register SSS to the address in register MMM. Note: SSS=MMM=4,5,6 or 7 not supported. Not interruptible. | |
| PSHR | SSS | 9 | | | 1 | 001 | 110 | SSS | PuSH data from Register SSS to the stack. Not interruptible. | |
| MVI@ | MMM, DDD | 8 / 11 | | | 1 | 010 | MMM | DDD | MoVe In data from the address in register MMM to register DDD. | |
| PULR | DDD | 11 | | | 1 | 010 | 110 | DDD | PULl data from the stack to Register DDD. | |
| ADD@ | MMM, DDD | 8 / 11 | | | 1 | 011 | MMM | DDD | ADD data located at the address in register MMM to contents of register DDD. Results to DDD. | S, Z, C, OV |
| SUB@ | MMM, DDD | 8 / 11 | | | 1 | 100 | MMM | DDD | SUBtract data located at the address in register MMM from contents of register DDD. Results to DDD. | S, Z, C, OV |
| CMP@ | MMM, DDD | 8 / 11 | | | 1 | 101 | MMM | SSS | CoMPare data located at the address in register MMM with contents of register SSS by subtraction. Results not stored. | S, Z, C, OV |
| AND@ | MMM, DDD | 8 / 11 | | | 1 | 110 | MMM | DDD | logical AND contents of register DDD with data located at the address in register MMM. Results to DDD. | S, Z |
| XOR@ | MMM, DDD | 8 / 11 | | | 1 | 111 | MMM | DDD | eXclusive OR contents of regsister DDD with data located at the address in register MMM. Results to DDD. | S, Z |

NOTE: 8 / 11 - 11 Cycles if MMM = 6, 8 Cycles otherwise

## 3.6.10 IMMEDIATE DOUBLE BYTE DATA — REGISTER

| Mnemonic | Operand | Cycles | Instruction | | | | | | Description | Status Change |
|---|---|---|---|---|---|---|---|---|---|---|
| SDBD | | | X | XXX | XX0 | 000 | 000 | 001 | MoVe In Immediate double byte data to register DDD. | |
| MVII | I-I, DDD | 14 | X | XXX | XX1 | 010 | 111 | DDD | | |
| | | | X | XXX | XXX | XLL | LLL | LLL | | |
| | | | X | XXX | XXX | XUU | UUU | UUU | | |
| SDBD | | | | | 0 | 000 | 000 | 001 | ADD Immediate double byte data to contents of register DDD. | S, Z, C, OV |
| ADDI | I-I, DDD | 14 | | | 1 | 011 | 111 | DDD | Results to DDD. | |
| | | | | | | LL | LLL | LLL | | |
| | | | | | | UU | UUU | UUU | | |
| SDBD | | | | | 0 | 000 | 000 | 001 | SUBtract Immediate double byte data from contents of register | S, Z, C, OV |
| SUBI | I-I, DDD | 14 | | | 1 | 100 | 111 | DDD | DDD. Results to DDD. | |
| | | | | | | LL | LLL | LLL | | |
| | | | | | | UU | UUU | UUU | | |
| SDBD | | | | | 0 | 000 | 000 | 001 | CoMPare Immediate double byte data with contents of register | S, Z, C, OV |
| CMPI | I-I, DDD | 14 | | | 1 | 101 | 111 | DDD | SSS by subtraction. Results not stored. | |
| | | | | | | LL | LLL | LLL | | |
| | | | | | | UU | UUU | UUU | | |
| SDBD | | | | | 0 | 000 | 000 | 001 | logical AND Immediate double byte data with contents of register | S, Z |
| ANDI | I-I, DDD | 14 | | | 1 | 100 | 111 | DDD | DDD. Results to register DDD. | |
| | | | | | | LL | LLL | LLL | | |
| | | | | | | UU | UUU | UUU | | |
| SDBD | | | | | 0 | 000 | 000 | 001 | eXclusive OR Immediate double byte data with contents of | S, Z |
| XORI | I-I, DDD | 14 | | | 1 | 101 | 111 | DDD | register DDD. Results to register DDD. | |
| | | | | | | LL | LLL | LLL | | |
| | | | | | | UU | UUU | UUU | | |

NOTE: I - I — UUUUUUUULLLLLLLL; L - L indicates low byte of literal; U - U indicates upper byte.
NOTE: The SDBD instruction is normally supplied by the assembler as required to properly generate machine code.

## 3.6.11 INDIRECT ADDRESSED DOUBLE BYTE DATA — REGISTER

| Mnemonic | Operand | Cycles | Instruction | | | | | | Description | Status Change |
|---|---|---|---|---|---|---|---|---|---|---|
| SDBD | | | X | XXX | XX0 | 000 | 000 | 001 | MoVe In double byte data from the address in register MMM to | |
| MVI@ | MMM, DDD | 14 | X | XXX | XX1 | 010 | MMM | DDD | register DDD. | |
| SDBD | | | | | 0 | 000 | 000 | 001 | ADD double byte data located at the address in register MMM to | S, Z, C, OV |
| ADD@ | MMM, DDD | 14 | | | 1 | 011 | MMM | DDD | contents of register DDD. Results to DDD. | |
| SDBD | | | | | 0 | 000 | 000 | 001 | SUBtract double byte data located at the address in register MMM | S, Z, C, OV |
| SUB@ | MMM, DDD | 14 | | | 1 | 100 | 111 | DDD | from contents of register DDD. Results to DDD. | |
| SDBD | | | | | 0 | 000 | 000 | 001 | CoMPare double byte data located at the address in register | S, Z, C, OV |
| CMP@ | MMM, DDD | 14 | | | 1 | 101 | 111 | DDD | MMM with the contents of register SSS by subtraction. Results | |
| | | | | | | | | | not stored. | |
| SDBD | | | | | 0 | 000 | 000 | 001 | logical AND double byte data located at the address in register | S, Z |
| AND@ | MMM, DDD | 14 | | | 1 | 100 | 111 | DDD | MMM with contents of register DDD. Results to DDD. | |
| SDBD | | | | | 0 | 000 | 000 | 001 | eXclusive OR double byte data located at the address in register | S, Z |
| XOR@ | MMM, DDD | 14 | | | 1 | 101 | 111 | DDD | MMM with contents of regsiter DDD. Results to DDD. | |

## 3.6.12 SYMBOLIC NOTATION — The following symbolic notation is used in all CP1600 instruction documentation.

| | | | | | | |
|---|---|---|---|---|---|---|
| **Address** | | | **Functions** | + | - addition | |
| **Modes:** | R | - register | | - | - subtraction | |
| | blank' | - direct address | | \| | - inclusive OR | |
| | I | - immediate data | | ^ | - exclusive OR | |
| | @ | indirect address | | & | - AND | |
| | | | | ( ) | - contents of | |
| **Operands:** | SSS | - Source Register | | <- | - is replaced by | |
| | DDD | - Destination Register | | < > | - optional operanc | |
| | MMM | - Register Address Mode | **Status:** | S | - Sign bit | |
| | RR | - Register (0 - 3) | | Z | - Zero bit | |
| | N | - Number of Shifts (0 = 1, 1 = 2) | | C | - Carry bit | |
| | S | - Sign of Address Displacement | | OV | - Overflow bit | |
| | EEEE | - External Condition Code (0 - 15) | | | | |
| | DA | - Destination Address | | | | |
| | SA | - Source Address | | | | |
| | I - I | - Immediate data word | | | | |
| | P - P | - Address displacement for Branch | | | | |

| **MMM** | **- Register Address Mode** |
|---|---|
| 000 | - direct address in location following instruction |
| 001 | - indirect address for Register 1 |
| 010 | - indirect address for Register 2 |
| 011 | - indirect address for Register 3 |
| 100 | - indirect address for Register 4, post increment |
| 101 | - indirect address for Register 5, post increment |
| 110 | - indirect address for Register 6, post increment for MVO only; indirect address for Register 6, pre decrement for all instructions except MVC |
| 111 | - indirect address for Register 7, post increment. (Immediate data in location following instruction.) |