# IBM® OPTIMUM PROGRAM CHART—650 DATA PROCESSING SYSTEM

| | Location of Inst. $n, n_e$ | Operation | Data | | Instruction |
|---|---|---|---|---|---|
| ARITHMETIC CODES | n | Add-Subtract:10, 11,15,16,17,18,60, 61,65,66,67,68... | n+3 | EVEN | d+5 |
| | | | | ODD | d+4 |
| | n | MPY(19) | n+3 | EVEN | d+21+2Σm |
| | | | | ODD | d+20+2Σm |
| | n | DIV(14),DVR(64) | n+3 | EVEN | d+61+2Σq |
| | | | | ODD | d+60+2Σq |
| BRANCHING CODES | EVEN ODD | NZU(44) | n+3 n+4 | | n+4 n+5 |
| | EVEN ODD | NZE(45) | n+4 n+3 | | n+5 n+4 |
| | n | BMI(46) | n+3 | | n+4 |
| | n | BOV(47) | n+3 | | n+5 |
| | n | BD0(90), BD9(99) | n+4 | | n+5 |
| | n | BD1-BD8 (91-98) | n+3 | | n+5 |
| | n | NTS(25), NEF(54) | n+4 | | n+5 |
| | n | NZA-B-C (40,42,48) BMA-B-C (41,43,49) | n+3 | | n+4 |
| | n | BIN(26) | | | n+4 |
| MISCELLANEOUS OPERATION CODES | n | LDD(69), STD(24) | n+3 | | d+3 |
| | n | TLU(84) | n+3 | EVEN | A+5 |
| | | | | ODD | A+6 |
| | n | NOP(00), HLT(01) | | | n+4 |
| | EVEN ODD | STL(20) | n+5 n+4 | | d+3 |
| | EVEN ODD | STU(21) | n+4 n+5 | | d+3 |
| | EVEN ODD | SDA(22), SIA(23) | n+3 n+4 | | d+3 |
| | EVEN ODD | SRT(30), SLT(35), SCT(36), S=0 | | | n+6 n+5 |
| | EVEN ODD | SRT(30), SLT(35), SCT(36), S=1,2 | | | n+7 n+6 |
| | EVEN ODD | SRT(30), SLT(35), SCT(36), S=3-9 | | | n+7 thru (2S+3) n+6 thru (2S+2) |
| | EVEN ODD | SCT(36) S=10 | ** | | n+7 thru (2S+3) n+6 thru (2S+2) |
| | EVEN ODD | SRD(31) S=1-10 | | | n+7 thru (2S+5) n+6 thru (2S+4) |
| INDEXING REG. CODES | n | Add-Subtract - 50, 51,52,53,58,59,80, 81,82,83,88,89... | 0000-1999 | | n+5* |
| | n | " | 9000-9059 | | n+7* |
| | n | " | 8000 | | n+7* |
| | n | " | 8001 | | n+5* |
| | EVEN ODD | " | 8002 | | n+8* n+7* |
| | EVEN ODD | " | 8003 | | n+7* n+8* |
| | n | NZA-B-C (40,42,48) BMA-B-C (41,43,49) | n+3 | | n+4 |

Σm: Sum of Multiplier Digits
Σq: Sum of Quotient Digits
S: Number of Shifts
W: Number of words moved

**NOTATIONS:**
A: Location of found argument (TLU)
d: Data Address of Instruction
i: Instruction Address of Instruction

*Add one to the "I" Address if a complement cycle is taken.
**Where a Low-High limit is indicated for an "I" Address, the following instruction should be optimized from the High limit. This applies only when the "I" Address is placed between the specified limits.
# Add two if a complement cycle is taken.
$I_1$ Instruction address specified by the Indexing Register. The times shown include the execution of the NOP(00) instruction which follows an "I" Address of 8005-6-7.

| | Location of Inst. $n, n_e$ | Operation | Data | Instruction |
|---|---|---|---|---|
| USING 800X AND 90XX ADDRESSES | EVEN ODD | All Add-Subtract (Accumulators) | 8000, 8003 8005,8006,8007, 90XX | n+7 n+8 |
| | EVEN ODD | " | 8001 | n+7 n+6 |
| | EVEN ODD | " | 8002 | n+9 n+8 |
| | n | LDD(69) | 8000, 8005-6-7, 90XX | n+6 |
| | EVEN ODD | LDD(69) | 8002 | n+7 n+6 |
| | EVEN ODD | LDD(69) | 8003 | n+6 n+7 |
| | | All Add-Subtract (Accumulators) | EVEN ODD | (8000,8001, 8003, 90XX) d+5# d+4# |
| | | " | EVEN ODD | (8002) d+6# d+5# |
| | | " | EVEN ODD | (8005-6-7) $I_1$=d+9# $I_1$=d+8# |
| | | LDD(69) | d | (8000,8001,90XX) d+3 |
| | | " | EVEN ODD | (8002) d+4 d+3 |
| | | " | EVEN ODD | (8003) d+3 d+4 |
| | | " | d | (8005-6-7) $I_1$=d+7 |
| IAS | n | LIB(08), LDI(09) SIB(28), STI(29) | n+3 | d+W+2 |
| | n | SET(27) | | n+5 |
| TAPE | n | RTN(04), RTA(05), WTN(06), WTA(07), RTC(03), RWD(55), WTM(56), BST(57)... | | n+5 |
| | n | NTS(25), NEF(54) | n+4 | n+5 |
| DISK STOR. | n | SDS(85), RDS(86), WDS(87) | | n+5 |
| | n | BIN(26) | | n+4 |
| FLOATING DECIMAL | n | UFA(02) | n+3 | EVEN d+5 thru 33** ODD d+4 thru 32** |
| | n | FAD(32), FSB(33) FAM(37), FSM(38) | n+3 | EVEN d+5 thru 59** ODD d+4 thru 58** |
| | n | FMP(39) | n+3 | EVEN d+5 thru 47** ODD d+4 thru 46** |
| | n | FDV(34) | n+3 | EVEN d+5 thru 41** ODD d+4 thru 40** |

**ADDR. MOD. IND. REG.**

When optimizing the addresses of a "tagged" instruction the programmer must mentally adjust the Location of Instruction (n) to an Effective Location of Instruction ($n_e$) before adding the optimizing factors. The optimizing of a tagged instruction is always done from the effective location ($n_e$) and this can change the even-odd word consideration. With one address tagged, ($n_e$) = n + 1. With both addresses tagged, ($n_e$) = n + 2.

## ARITHMETIC CODES

| | | |
|---|---|---|
| Add Upper | 10 | AUP |
| Add Lower | 15 | ALO |
| Subtract Upper | 11 | SUP |
| Subtract Lower | 16 | SLO |
| Multiply | 19 | MPY |
| Divide | 14 | DIV |
| Divide Reset Upper | 64 | DVR |
| Reset Add Upper | 60 | RAU |
| Reset Add Lower | 65 | RAL |
| Reset Subtract Upper | 61 | RSU |
| Reset Subtract Lower | 66 | RSL |
| Add Absolute (Magnitude) to Lower | 17 | AML |
| Subtract Absolute (Magnitude) from Lower | 18 | SML |
| Reset Add Absolute (Magnitude) to Lower | 67 | RAM |
| Reset Subtract Absolute (Magnitude) from Lower | 68 | RSM |

## INDEXING REGISTER CODES

### REGISTER "A"

| | | |
|---|---|---|
| Add | 50 | AXA |
| Subtract | 51 | SXA |
| Reset Add | 80 | RAA |
| Reset Subtract | 81 | RSA |
| Branch Non-Zero | 40 | NZA |
| Branch Minus | 41 | BMA |

### REGISTER "B"

| | | |
|---|---|---|
| Add | 52 | AXB |
| Subtract | 53 | SXB |
| Reset Add | 82 | RAB |
| Reset Subtract | 83 | RSB |
| Branch Non-Zero | 42 | NZB |
| Branch Minus | 43 | BMB |

### REGISTER "C"

| | | |
|---|---|---|
| Add | 58 | AXC |
| Subtract | 59 | SXC |
| Reset Add | 88 | RAC |
| Reset Subtract | 89 | RSC |
| Branch Non-Zero | 48 | NZC |
| Branch Minus | 49 | BMC |

## BRANCHING CODES

### ACCUMULATOR

| | | |
|---|---|---|
| Branch Non-Zero Upper | 44 | NZU |
| Branch Non-Zero | 45 | NZE |

## BRANCHING CODES CONT'D

| | | |
|---|---|---|
| Branch Minus | 46 | BMI |
| Branch on Overflow | 47 | BOV |

### DISTRIBUTOR

| | | |
|---|---|---|
| Branch on 8 in Position 10 | 90 | BDO |
| Branch on 8 in Position 1 | 91 | BD1 |
| Branch on 8 in Positions 2-8 | 92-98 | BD2-8 |
| Branch on 8 in Position 9 | 99 | BD9 |

### INDEXING REGISTER "A"

| | | |
|---|---|---|
| Branch Non-Zero | 40 | NZA |
| Branch Minus | 41 | BMA |

### INDEXING REGISTER "B"

| | | |
|---|---|---|
| Branch Non-Zero | 42 | NZB |
| Branch Minus | 43 | BMB |

### INDEXING REGISTER "C"

| | | |
|---|---|---|
| Branch Non-Zero | 48 | NZC |
| Branch Minus | 49 | BMC |

### TAPE—RAMAC

| | | |
|---|---|---|
| Branch No Tape Signal | 25 | NTS |
| Branch No End of File | 54 | NEF |
| Branch on Inquiry | 26 | BIN |

## INPUT—OUTPUT CODES

### SYNCHRONIZER 1

| | | |
|---|---|---|
| Read | 70 | RD1 |
| Write | 71 | WR1 |
| Read Conditional | 72 | RC1 |

### SYNCHRONIZER 2

| | | |
|---|---|---|
| Read | 73 | RD2 |
| Write | 74 | WR2 |
| Read Conditional | 75 | RC2 |

### SYNCHRONIZER 3

| | | |
|---|---|---|
| Read | 76 | RD3 |
| Write | 77 | WR3 |
| Read Conditional | 78 | RC3 |

## IMMEDIATE ACCESS STORAGE

| | | |
|---|---|---|
| Load IAS Block | 08 | LIB |
| Load IAS | 09 | LDI |
| Store IAS Block | 28 | SIB |
| Store IAS | 29 | STI |
| Set IAS Timing Ring | 27 | SET |

## MISC. OPERATION CODES

### STORE

| | | |
|---|---|---|
| Store Lower Accumulator | 20 | STL |
| Store Upper Accumulator | 21 | STU |
| Store "D" Address of Lower Accumulator | 22 | SDA |
| Store "I" Address of Lower Accumulator | 23 | SIA |
| Store Distributor | 24 | STD |

### SHIFT

| | | |
|---|---|---|
| Shift Right | 30 | SRT |
| Shift Right and Round | 31 | SRD |
| Shift Left | 35 | SLT |
| Shift Left and Count | 36 | SCT |

### MISC.

| | | |
|---|---|---|
| No Operation | 00 | NOP |
| Stop (Halt) | 01 | HLT |
| Load Distributor | 69 | LDD |
| Table Look-Up | 84 | TLU |

## TAPE

| | | |
|---|---|---|
| Read Tape Numeric | 04 | RTN |
| Read Tape Alphameric | 05 | RTA |
| Write Tape Numeric | 06 | WTN |
| Write Tape Alphameric | 07 | WTA |
| Read Tape for Checking | 03 | RTC |
| Branch No Tape Signal | 25 | NTS |
| Branch No End of File | 54 | NEF |
| Rewind Tape | 55 | RWD |
| Write Tape Mark | 56 | WTM |
| Backspace Tape | 57 | BST |

## RAMAC®—DISK STORAGE

| | | |
|---|---|---|
| Seek Disk Storage | 85 | SDS |
| Read Disk Storage | 86 | RDS |
| Write Disk Storage | 87 | WDS |
| Branch on Inquiry | 26 | BIN |
| Reply to Inquiry | 79 | RPY |

## FLOATING DECIMAL ARITH.

| | | |
|---|---|---|
| Floating Add | 32 | FAD |
| Floating Subtract | 33 | FSB |
| Floating Multiply | 39 | FMP |
| Floating Divide | 34 | FDV |
| Unnormalized Floating Add | 02 | UFA |
| Floating Add Absolute (Magnitude) | 37 | FAM |
| Floating Subtract Absolute (Magnitude) | 38 | FSM |

OPTIMUM PROGRAM CHART
IBM 650 DATA PROCESSING SYSTEM

Form 22-6219-5 Printed in U.S.A.

84.17