DATATRON 205
COMMAND LIST

STANDARD SYMBOLS

xxxx— four digit address of storage cell on magnetic drum, or four digit address of block of information on magnetic tape.

p — control digit inserted in command word to act as breakpoint instruction.

f — control digit inserted in command word to act as format instruction to output device.

n — control digit inserted in command word to designate quantity.

0 — digit normally used to complete word.

u — control digit inserted in command word to designate unit number of input or output component.

h — control digit inserted in command word to designate head number for magnetic tape search operation.

t — control digit inserted in command word to set punch (or printer) relays.

ARITHMETIC

▼ ADDITION

CAD   **CLEAR, ADD**

000p 64 xxxx

Clear the A Register. Add the contents of xxxx.

CADA   **CLEAR, ADD ABSOLUTE**

000p 66 xxxx

Clear the A Register. Add the absolute value of the contents of xxxx.

AD   **ADD**

000p 74 xxxx

Add the contents of xxxx to the contents of the A Register.

ADA   **ADD ABSOLUTE**

000p 76 xxxx

Add the absolute value of the contents of xxxx to the contents of the A Register.

FAD   **FLOATING ADD**

000p 80 xxxx

Add the floating point number in xxxx to the floating point number in the A Register.

DAD   **DIGIT ADD**

0000 10 0000

Stop machine operation. Add the next digit read (from manual keyboard or paper tape reader) to the least significant position of the A Register.

▼ SUBTRACTION

CSU   **CLEAR, SUBTRACT**

000p 65 xxxx

Clear the A Register. Subtract the contents of xxxx.

CSUA   **CLEAR, SUBTRACT ABSOLUTE**

000p 67 xxxx

Clear the A Register. Subtract the absolute value of the contents of xxxx.

SU   **SUBTRACT**

000p 75 xxxx

Subtract the contents of xxxx from the contents of the A Register.

SUA   **SUBTRACT ABSOLUTE**

000p 77 xxxx

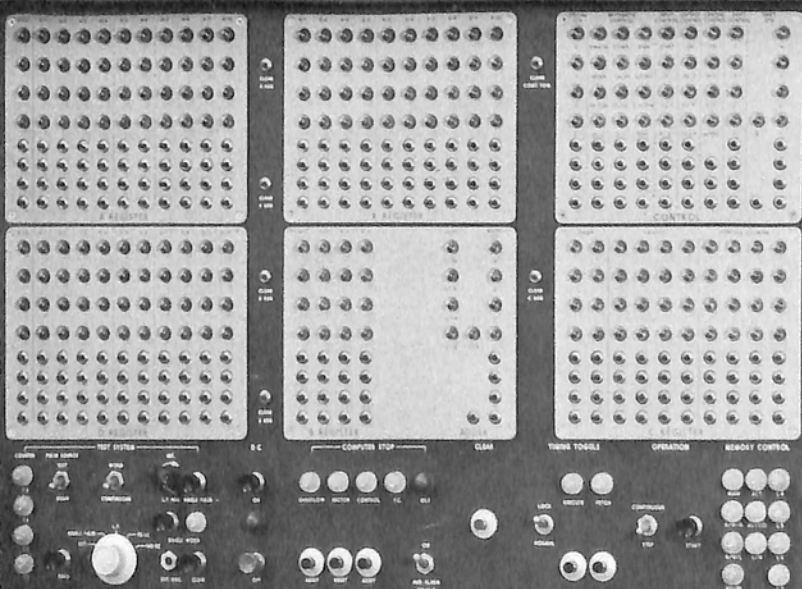Subtract the absolute value of the contents of xxxx from the contents of the A Register.

FSU   **FLOATING SUBTRACT**

000p 81 xxxx

Subtract the floating point number in xxxx from the floating point number in the A Register.

▼ MULTIPLICATION

M   **MULTIPLY**

000p 60 xxxx

Multiply the contents of **xxxx** by the contents of the A Register. Insert the twenty digit product in the A Register and the R Register. The most significant digits are in the A Register.

MRO   **MULTIPLY, ROUND**

000p 70 xxxx

Multiply the contents of **xxxx** by the contents of the A Register. Round the product to ten digits. Clear the R Register.

FM   **FLOATING MULTIPLY**

000p 82 xxxx

Multiply the floating point number in **xxxx** by the floating point number in the A Register. Insert the eighteen digit floating point product in the A Register and the R Register. The most significant digits are in the A Register.

▼ DIVISION

DIV   **DIVIDE**

000p 61 xxxx

Divide the twenty digit contents of the A Register and the R Register by the contents of **xxxx**.

   (a) If Overflow indicates ON, clear the A Register and the R Register.

   (b) If Overflow indicates OFF, insert the quotient in the A Register, and insert the undivided remainder (if any) in the R Register.

FDIV   **FLOATING DIVIDE**

000p 83 xxxx

Divide the eighteen digit floating point number in the A Register and the R Register by the floating point number in **xxxx**. Insert the ten digit floating point quotient in the A Register. Insert the undivided remainder (if any) in the least significant positions of the R Register.

USING THE B REGISTER

SB   **SET B**

000p 72 xxxx

Set the B Register to the value of the four least significant positions of **xxxx**.

BA   **B TO A**

000p 11 0000

Clear the A Register. Add the contents of the B Register.

IB   **INCREASE B**

000p 32 0000

Add one to the contents of the B Register.

DB   **DECREASE B**

000p 22 xxxx

Subtract one from the contents of the B Register.
(a) If the new B Register setting is 9999 (0000 − 1), control continues in sequence.
(b) If the new B Register setting is not 9999, change control to **xxxx**.

MANIPULATION AND TRANSFER OF INFORMATION

ST   **STORE**

000p 12 xxxx

Store the contents of the A Register in **xxxx**.

STC   **STORE, CLEAR**

000p 02 xxxx

Store the contents of the A Register in **xxxx**. Clear the A Register.

SL   **SHIFT LEFT**

000p 14 00nn

Shift the contents of the A Register and the R Register **nn** places left. The **nn** digits shifted out of the left end of the A Register re-enter the right end of the R Register in the same order. The sign does not move.

SR   **SHIFT RIGHT**

000p 13 00nn

Shift the contents of the A Register and the R Register **nn** places right. The **nn** digits shifted out of the right end of the R Register are lost, and **nn** zeros enter the left end of the A Register. The sign does not move. The maximum value for **nn** is 19.

NOR   **NORMALIZE (CHANGE ON ZERO)**

000p 15 xxxx

See definition under "Decision Making and Branching" commands.

CIRA   **CIRCULATE A**

000p 01 00nn

Shift the contents (including sign) of the A Register **nn + 1** places left. The digits shifted out of the left end of the A Register re-enter the right end of the A Register in the same order.

EX   **EXTRACT**

000p 63 xxxx

Extract from the contents of the A Register by changing each digit in the A Register (including sign) to zero if the digit in the corresponding position in **xxxx** is zero. The digit in the A Register remains unchanged if the digit in the corresponding position in **xxxx** is one.

CR   **CLEAR R**

000p 33 0000

Clear the R Register.

RO   **ROUND**

000p 23 0000

Round the twenty digit contents of the A Register and the R Register to ten digits. Clear the R Register.

BT4   **BLOCK TO LOOP 4**

000p 34 xxxx

Block transfer the contents of twenty consecutive main storage cells, beginning with **xxxx**, to the 4000 quick access loop. Use **BT5 (35)** for the 5000 loop, **BT6 (36)** for the 6000 loop, and **BT7 (37)** for the 7000 loop.

BF4   **BLOCK FROM LOOP 4**

000p 24 xxxx

Block transfer the contents of the 4000 quick access loop to twenty consecutive main storage cells, beginning with **xxxx**. Use **BF5 (25)** for the 5000 loop, **BF6 (26)** for the 6000 loop, and **BF7 (27)** for the 7000 loop.

UA   **UNIT ADJUST**

000p 06 0000

Increase by one the most significant position of the A Register if the digit in this position is even.

ADSC   **ADD SPECIAL COUNTER**

000p 16 0000

Add the contents of the Special Counter to the least significant position of the A Register.

SUSC   **SUBTRACT SPECIAL COUNTER**

000p 17 0000

Subtract the contents of the Special Counter from the least significant position of the A Register.

DECISION MAKING AND BRANCHING

STOP   **STOP**

000p 08 0000

Stop machine operation.

OSGD   **OVERFLOW ON SIGN DIFFERENCE**

000p 73 xxxx

If the sign of the A Register differs from the sign of **xxxx**, Overflow indicates ON.

CNZ   **CHANGE ON NON-ZERO**

000p 04 xxxx

Test the contents of the A Register (not the sign) for zero.
(a) If the A Register setting is zero, set the sign of the A Register to zero and continue control in sequence.
(b) If the A Register setting is not zero, change control to **xxxx**.

NOR   **NORMALIZE (CHANGE ON ZERO)**

000p 15 xxxx

(a) If the content of the A Register is not zero, shift the twenty digits in the A Register and the R Register left until the most significant position in the A Register is not zero. The sign does not move. Record the number of shifts in the Special Counter.
(b) If the content of the A Register is zero, shift the contents of the R Register left into the A Register, clear the R Register, and change control to **xxxx**. The sign does not move.

CC   **CHANGE CONDITIONALLY**

000p 28 xxxx

Overflow indicates ON: Change control to **xxxx**. Reset Overflow.
Overflow indicates OFF: Control continues in sequence.

CCB   **CHANGE CONDITIONALLY, BLOCK**

000p 38 xxxx

Overflow indicates ON: Block transfer the contents of twenty consecutive main storage cells, beginning with **xxxx**, to the 7000 loop. Change control to **70xx**. Reset Overflow.
Overflow indicates OFF: Control continues in sequence.

CCR   **CHANGE CONDITIONALLY, RECORD**

000p 29 xxxx

Overflow indicates ON: Clear the R Register. Store in the four most significant positions of the R Register the address (as contained in the Control Counter) of the command next in sequence. Change control to **xxxx**. Reset Overflow.
Overflow indicates OFF: Control continues in sequence.

CCBR   **CHANGE CONDITIONALLY, BLOCK, RECORD**

000p 39 xxxx

Overflow indicates ON: Block transfer the contents of twenty consecutive main storage cells, beginning with **xxxx**, to the 7000 loop. Clear the R Register. Store in the four most significant positions of the R Register the address (as contained in the Control Counter) of the command next in sequence. Change control to **70xx**. Reset Overflow.
Overflow indicates OFF: Control continues in sequence.

CU   **CHANGE UNCONDITIONALLY**

000p 20 xxxx

Change control to **xxxx**.

CUB   **CHANGE UNCONDITIONALLY, BLOCK**

000p 30 xxxx

Block transfer the contents of twenty consecutive main storage cells, beginning with **xxxx**, to the 7000 loop. Change control to **70xx**.

CUR   **CHANGE UNCONDITIONALLY, RECORD**

000p 21 xxxx

Clear the R Register. Store in the four most significant positions of the R Register the address (as contained in the Control Counter) of the command next in sequence. Change control to **xxxx**.

CUBR   **CHANGE UNCONDITIONALLY, BLOCK, RECORD**

000p 31 xxxx

Block transfer the contents of twenty consecutive main storage cells, beginning with **xxxx**, to the 7000 loop. Clear the R Register. Store in the four most significant positions of the R Register the address (as contained in the Control Counter) of the command next in sequence. Change control to **70xx**.

INPUT-OUTPUT

▼ TYPEWRITER

PTW   **WRITE**

000p 03 ffnn

Write on typewriter, transferring the sign and **nn** digits from the A Register. Digits **ff** act as an instruction to the typewriter. Shift the contents (including sign) of the A Register **nn + 1** places left. The digits shifted out of the left end of the A Register re-enter the right end of the A Register in the same order.

PTWF   **WRITE FORMAT**

000p 07 0f00

Actuate typewriter as directed by digit **f**.

▼ PAPER TAPE

PTR   READ

0000 00 xxxx

Read from paper tape, transferring words to consecutive storage cells on the drum starting with **xxxx**. Stop input and start computation after reading a **CU, CUB, CUR,** or **CUBR** command (with a 6 or 7 in the sign position).

PTW   WRITE

000p 03 ffnn

Punch on paper tape, transferring the sign and **nn** digits from the A Register. Punch digits **ff** on tape to act as an instruction to a typewriter. Shift the contents (including sign) of the A Register **nn + 1** places left. The digits shifted out of the left end of the A Register re-enter the right end of the A Register in the same order.

PTWF   WRITE FORMAT

000p 07 0f00

Punch the digit **f** on paper tape to act as an instruction to a typewriter.

▼ MAGNETIC TAPE

MTRW   REWIND
00up 52 0000
Rewind DataReader **u**.

MTS   SEARCH
Ohup 42 xxxx
Search for block **xxxx** under head **h** on DataReader **u**. Overflow indicates ON if a previous **MTS** command has not been completed.

MTR   READ
nnup 40 xxxx
Read **nn** consecutive blocks of twenty words each from DataReader **u**, transferring words to consecutive storage cells on the drum starting with **xxxx**. Overflow indicates ON if a previous **MTS** command has not been completed.

MTW   WRITE
nnup 50 xxxx
Write **nn** consecutive blocks of twenty words each on DataReader **u**, transferring words from consecutive storage cells on the drum starting with **xxxx**. Overflow indicates ON if a previous **MTS** command has not been completed.

▼ CARD FEED, CARD PUNCH AND TABULATOR WITH MODEL 500 CONVERTER

CDR   READ

nnnp 44 xxxx
Read **1000 − nnn** cards continuously, transferring words to consecutive storage cells on the drum starting with **xxxx**.

CDW   WRITE

nnnp 54 xxxx

Punch **1000 − nnn** cards (or print **1000 − nnn** lines) continuously, transferring words from consecutive storage cells on the drum starting with **xxxx**.

EXC   EXTERNAL CONTROL

000p 71 xxxx

Insert the contents of **xxxx** in the D Register. For each of the eight most significant digits in the D Register there is an electronic switch. A "3" changes the state of the corresponding switch, a "2" closes the corresponding switch, a "1" opens the corresponding switch, and a "0" does not alter the state of the corresponding switch.

▼ CARD FEED, CARD PUNCH AND TABULATOR WITH CARDATRON

CDRF   READ FORMAT

Ofup 48 xxxx

Load format band **f** on input **u**, transferring words from consecutive storage cells on the drum starting with **xxxx**.

CDWF   WRITE FORMAT

Ofup 58 xxxx

Load format band **f** on output **u**, transferring words from consecutive storage cells on the drum starting with **xxxx**.

CDR   READ

00up 44 xxxx

Read the contents of one card from input **u**, transferring words to consecutive storage cells on the drum starting with **xxxx**. Reload input **u** with the contents of the next card.

CDW   WRITE

tfup 54 xxxx

Punch one card (or print one line) at output **u**, transferring words from consecutive storage cells on the drum starting with **xxxx**. Edit the information as directed by format band **f**. Control the punch (or printer) as directed by digit **t**.

CDRI   READ INTERROGATE

00up 45 xxxx

Interrogate input **u**. If input **u** is ready to read, clear the R Register. Store in the four most significant positions of the R Register the address (as contained in the Control Counter) of the command next in sequence. Change control to **xxxx**. If input **u** is not ready to read, control continues in sequence.

CDWI   WRITE INTERROGATE

00up 55 xxxx

Interrogate output **u**. If output **u** is ready to write, clear the R Register. Store in the four most significant positions of the R Register the address (as contained in the Control Counter) of the command next in sequence. Change control to **xxxx**. If output **u** is not ready to write, control continues in sequence.