

## BTM 1200 series: instruction set(s) and instruction times.

### Contents.

1. Instruction layout and central registers.
2. Instruction set.
3. Input/output details.
4. Instruction times.

### 1. Instruction layout and central registers.

The format of a 40-bit instruction is as follows (see reference 2 in section T1X5):

13 bits		5 bits		13 bits		1	1	7 bits	
<i>Operand address</i>		<i>Function</i>		<i>Next instr. address</i>		<i>TD</i>	<i>H</i>	<i>Shift count</i>	
1	13	14	18	19	31			34	40

The bits of the *operand address* are interpreted as follows:

- Bit 1: take operand from Q register
- Bit 2: take operand from B register
- Bit 3: spare (unassigned)
- Bits 4 – 9: drum track number
- Bits 10 – 13: word within the track.

The bits of the *next-instruction address* are interpreted as follows:

- Bit 19: take next instruction from Q register
- Bit 20: take next instruction from B register
- Bit 21: spare (unassigned)
- Bits 22 – 27: drum track number
- Bits 28 – 31: word within the track.

TD in the instruction is the *track delay* bit. If set, this inhibits access to the drum for four word times (5 milliseconds) to allow for the track-selection relays to operate. This safeguard is necessary during address-modification where the exact location of the next instruction is not known at the time of writing the program.

H is a *Halt* bit. If set, and if a switch is also set on the operator's console, then the program will stop at this instruction.

The seven *shift count* bits give a value denoted as n. This value is used for a variety of purposes, for example to specify:

- (a) the number of places n to be shifted in instructions nTA, nRA, etc.;
- (b) the number of times that the additions in instructions BCA, GA are repeated;
- (c) the number of bits in a record, for the instructions nV, nW, nY and nX;
- (d) the number and position of digits to be tested in the conditional branch instructions nJ, nK and nE;
- (e) the form and location of information to be output in the P instruction.

The BTM 1201's central processing unit contains four registers visible to programmers. Each is a 40-bit shift register. Their abbreviated names are as follows:

A	accumulator
Q	temporary working space
M	multiplier register and temporary working space
B	temporary working space.

## 2. Instruction set.

There are 32 functions (op codes), each of which was given an alphabetical abbreviation.

CA	clear the accumulator A and add in the operand
A	add the operand to the quantity in the accumulator A
CS	clear and subtract
S	subtract
BCA	block clear and add a sequence of n words, starting at the operand address
GA	group accumulate: as above, but add the words cyclically in turn to the M, B, Q and A registers.
TM	transfer an operand to the M register – (ie, <i>load M</i> )
TB	transfer an operand to the M register – (ie, <i>load B</i> )
TQ	transfer an operand to the M register – (ie, <i>load Q</i> )
nV	move a record consisting of the first n digits or last n digits ( $0 \leq n \leq 40$ ) from A to the drum. The value in the <i>shift count</i> determines which bits are selected to form the record.
nW	move an n-digit record from the M register and clear the M register
nX	move an n-digit record from the Q register
nY	move an n-digit record from the B register
BR	move the contents of {M, Q, B, A} registers sequentially to the drum
nTA	shift A arithmetically right by n places – (the 'T' indicates that the 'true' sign of the value is propagated)
nRA	shift A circularly n places right – (the 'R' signifies 'round')
nT	similar to nTA but the M register holds any digits dropping off the end of A.
nR	similar to nRA but the M register holds any digits dropping off the end of A and then any digits dropping off the end of M are fed back into A.
nAQ	similar to nRA but the Q register also holds any digits dropping off the end of A
nTAB	similar to nTA but the Q register holds any digits dropping off the end of A
nRQ	similar to nRA but the Q register holds any digits dropping off the end of A
H	multiply: the multiplier is in M, the multiplicand is taken from either B or Q, and the most-significant 40 bits of the result is returned to the accumulator A. The multiplication sequence was based on the Booth Algorithm, which had the side-effect of leaving the least-significant half of the double-length result in the M register.
AH	add and multiply: the initial contents of A are added to the least significant 40 digits of the product.
D	divide
C	convert a binary number to groups of four binary-coded decimal (BCD) digits, which then represent decimal, sterling, etc. values – see below.
nJ	if $n^{\text{th}}$ digit of A is 1 then jump to the <i>operand address</i> as given in the

	instruction; otherwise, proceed as normal to the <i>next-instruction address</i> . The digit in A to be tested is specified by a value on the <i>shift count</i> .
nK	jump similarly for the n <sup>th</sup> digit of M
nE	test the most-significant n digits or the least-significant n digits of A. If at least one of the selected group is a 1, then jump to the <i>operand address</i> as given in the instruction. The value in the <i>shift count</i> determines whether the least-significant or most-significant n digits are to be tested.
G	test whether drum track 63, used as an output buffer, is in use or empty – (see also below).
F	feed a card in the card reader, in preparation to converting the incoming information to binary – (see below).
P	Initiate an output operation, which may be either a print or punch depending upon the value in <i>shift count</i> .
Z	halt the program – ie stop.

### 3. Input/output details.

Up to 50 columns of input information per card can be read in. The incoming information may be in decimal, or variable radix quantities such as sterling currency or avoirdupois weight; it is then converted in real time to binary. Each type of conversion requires a set of pre-defined binary constants to be stored at pre-determined tracks on the drum. Each card carries the track number of the relevant set of constants for the type of conversion required for that card. The conversion process uses the four central registers {A, Q, M, B} and four adders, so that a programmer should save the contents of these registers via a BR instruction before activating the conversion.

On output, the result of a calculation is first converted into a string of four-bit binary-coded decimal (BCD) groups. The BCD string can represent decimal, sterling, or other pre-defined notation. The BCD string is stored in a 64-digit output buffer at track 63 on the drum. Hardware then takes this BCD string and causes the relevant symbols to be either punched on a card or printed.

### 4. Instruction times.

Input/output was, for the mid-1950s, reasonably rapid. Input is via standard 80-column punched cards, at a rate of 125 cards/minute. Output is either via punched cards at about 100 cards/minute or via a line-at-a-time printer. The input/output facilities include hardware assistance for conversion to/from binary of values expressed in a variety of practical formats such as decimal, sterling, weight, etc. As far as can be determined, data conversion was carried out in 'real time' and allowed the peripheral equipment to transfer at full rate (eg 125 cards/minute for input or 100 cards/minute for output).

Since instructions and operands were stored on a sequential memory device (the magnetic drum), it is appropriate to quote minimum and maximum times for the arithmetic instructions. Here are two sample cases:

ADD time, in milliseconds:            minimum = 1.25; maximum = 21.25.  
MULTIPLY time, in milliseconds:    minimum = 3.75; maximum = 50.0.