A.T. Gough.

THE CITY AND GUILDS COMPUTER

INDEX

**** **** ****

The City & Guilds Computer.

The translator program simulates on the Elliott 903 the operation of the hypothetical City & Guilds computer.

(i) The Store

This computer has 1000 store locations addressed from 0, 1, 2.....999. Each location may hold either one instruction or a number which is stored in floating-point form to an accuracy of 7 significant decimal digits. The first ten locations have special applications.

Location	Function
0	The contents of this location will always be zero, and cannot be overwritten.
1	The address of the accumulator
2	Index registers used for modifying instruction addresses
3	
4	An index register, which is also used to store the return address link during a subroutine. (see JSR instruction).
5	Additional index registers.
6	
7	
8	
9	

(ii) The control register.

The control register contains the address of the instruction being obeyed. The register is automatically updated during the execution of the program ready for the extraction of the following instruction.

Programming the computer

The program consists of two types of order

(i) input directives which control the compiling process

(ii) program instructions, which are stored during compiling and are then obeyed.

Input Directives

These are obeyed during the compiling process and are not stored. They appear on separate lines in the program and are enclosed by brackets (TITLE)

As the program line following this directive is read into the computer, it will be copied on to the output tape and is not stored. The characters forming the title are preceded and followed by carriage return, line feed (cr lf).

If more than one line of text is required, (TITLE) must precede each line.

e.g. The input of

```
(TITLE)
PROGRAM EXAMPLE
(TITLE)
NUMBER ONE
```

will produce characters for the heading PROGRAM EXAMPLE

- blank line

NUMBER ONE

on the output tape. (Note the blank line separating the lines of text)

(STORE n) Is used before the first program instruction. n is the location where the instruction which follows is to be stored. (STORE 586) will place the first instruction in 586, the second in 587 etc. If it is desired to separate blocks of program, a STORE directive may be inserted for each group of instructions. Where several STORE directives are used, the locations to which they refer can be in any order.

(WAIT) causes a temporary stop during the input of the program tape. (WAIT) may be used where a program is continued on a second tape.

(EXECUTE n) is placed at the end of the program and indicates the location (n) of the first instruction which is to be obeyed.

Space, erase and blank tape characters are ignored during the input of a directive.

Program Instructions

Instructions are in the form

FCT n, m Q

Each instruction must be written on a new line. FCT the function part specifies the operation to take place.

n refers to any location in the range 0, ..., 999.

m is the modifier and refers to any one index register in the range 0, ..., 9.

Q is a query digit, which if omitted causes no further action. If Q is present on an instruction, it allows tracing of the program during running.

If the instruction contains a modifier then the comma must be used to separate n and m otherwise the comma should be omitted.

The operation of the modifier is to add the contents of the specified index register to the address n. e.g. If an instruction refers to address 501, 3 at a point where register 3 contains the number 29, the address to which access is gained will be 530.

In any store, a number is held in floating point form. During modification, the nearest integer to the contents of the index register is used. If, therefore, an index register contained 18.3, the modification would be by 18. -10.8 is taken as -11.

If the order contains no address part then the Q must be omitted. Space, erase and blank tape characters are ignored during the interpretation of an instruction.

Instruction Codes.

(In all these descriptions (n) should be read as "the contents of location n".)

Group 0 Instructions

All instructions in this group leave the store contents unchanged.

Mnemonic	Operation	Remarks
LDA n, m	(n+(m)) ⟶ A	Load operand into accumulator
ADD n, m	(A)+(n+(m)) ⟶ A	Add operand to accumulator
SUB n, m	(A)-(n+(m)) ⟶ A	Substract operand from accumulator
MLT n, m	(A)×(n-(m)) ⟶ A	Multiply accumulator by operand
DIV n, m	(A)/(n+(m)) ⟶ A	Divide accumulator by operand

Group 1

These instructions cause the address part of the instruction to be used directly as an operand rather than specifying the location number where the operand is to be found. n may be an integer in the range 0 to 999. The instructions may not be modified.

The integer must NOT be preceded by a + sign.

All instructions leave the contents of the store location unchanged.

Mnemonic	Operand	Remarks
LDAN n	n ⟶ A	Load integer n into accumulator
ADDN n	(A)+n ⟶ A	Add integer n to the accumulator contents
SUBN n	(A)-n ⟶ A	Subtract integer n from accumulator contents
MLTN n	(A)×n ⟶ A	Multiply accumulator contents by integer n.
DIVN n	(A)/n ⟶ A	Divide accumulator contents by integer n

e.g. LDAN 100 places the number 100 into the accumulator, and is not affected by the contents of store 100.

Group 2

Mnemonic	Operation	Remarks
STA n, m	A → (n+(m))	Copy accumulator into store location.

The contents of the store location are overwritten by the new value.

Group 3

Instructions in this group cause jumps to actual addresses; there is no symbolic addressing.

Mnemonic	Operation
JUN n, m	Jump unconditionally to n+(m)
JGR n, m	Jump to n+(m) if A≥0
JEQ n, m	Jump to n+(m) if A=0 Since numbers are stored in floating point form, the JEQ instruction may not operate as intended unless the value being compared with zero has been calculated from integers or exact binary fractions.
JSR n, m	Jump to subroutine in n+(m). The subroutine link is set in location 4. This is in the form of the address of the location following the JSR order. Exit from a subroutine is by the instruction JUN 0, 4 If one subroutine uses another, the link address must be removed from location 4 and stored in another index register.
JST n, m	Wait; jump to location n+(m) when the signal CONTINUE RUNNING is received. (See operating instructions)

Group 4

These are instructions to determine certain standard functions.

Mnemonic	Operation	Remarks
SQT n, m	$\sqrt{A} \longrightarrow A$	
	If the argument is negative then jump to n+(m)	
EXP n, m	exp (A) → A	
	If the argument is greater than 40 then jump to n+(m)	
LGN n, m	$\log_e(A) \rightarrow A$	
	If the argument is negative or zero then jump to n+(m)	
SIN n, m	sin(A) → A	The argument must be in radians.
COS n, m	cos(A) → A	The address part of these instructions is not used and may be omitted.
ARC n, m	arctan(A) → A	The result is in radians
ENT n, m	entier (A) → A	The integral part of (A) is placed in A. i.e. 8.6 becomes 8; -8.6 becomes -9.

Group 5.

This group of instructions contains the instructions for input and output of characters and numbers from paper tape.

Mnemonic	Operation	Remarks
RCT n, m	character → (n+(m))	Read a single character
PCT n, m	(n+(m)) → tape	Punch a single character on to tape.

It is not intended that a PCT instruction should be used except for the output of a character previously stored by RCT.

If the characters are input from a data tape which has contained numbers, then the first character must immediately follow the cr lf or sp sp which terminated the number.

If the characters are at the start of a data tape, then the tape must be placed in the reader so that the first character comes into the read position when the READ button is pressed.

PNL			Punched characters for new line (cr lf).
RNT	Number	acc	Read number from tape

Numbers read from tape may be in the form of

integers	123
decimals	123.0
floating point	1.23_{10}, 1230_{10}-1 etc.

and must be terminated by cr lf or sp sp. It is not necessary to precede positive numbers by a + sign. The length of the number should not exceed 7 decimal digits.

If an address is added to a PNL or RNT instruction, it will be ignored, except that tests are made for error routines 7 and 9.

PNT	n, m	(A)	tape	Punch the number in the accumulator onto tape

n and m specify the numbers of integral and fractional digits. Both n and m must be stated and if, for example no fractional digits are required the comma and zero must be included.

A negative number is preceded by - and a positive number by a space. Non significant leading zeros are replaced by spaces. All numbers are followed by sp sp. Including the decimal point, therefore, a number occupies (n+m+4) characters, except that an integer specified by (n,0) occupies n+3 characters.

If n+m is greater than 7, or if the setting 0,0 is used, the number will be printed in floating point form to 7 significant digits. This occupies 15 characters, including the final two spaces,

e.g. $-1.200000_{10}+01$ represents -12

If the number to be output is larger than the specified format, i.e. is $\geqslant 10^n$, then it will be output in floating point form.

Group 6.

Punch card orders are not included.

(v) Input of numbers with instructions.

Numbers may be input with the programme and are assigned to store in sequence with the instructions.

Each number must be preceded by a + or - sign; failure to insert the sign will produce error 2. The number must be terminated by cr lf.

The numbers may be in the form of integers, decimal numbers or in floating point, not exceeding 7 decimal digits.

(vi) Query digit

If the Query facility has been set (see operating instructions) then the following line of output is punched after each instruction containing a Q has been obeyed.

Q	Address of location	value of accumulator in floating point form.

Operating Instructions

The compiler program is issued as a sum-checked binary tape and may be run on an Elliott 903 computer fitted with 8192 words of store.

The compiler is read by Initial Instruction (8181)

The program should be punched on 8-channel tape on a Flexowriter Mode T or Teletype 33.

To TRANSLATE a mnemonic program, enter at 8

CONTINUE TRANSLATION after (WAIT) by entering at 14 or press INT

RUN the program after (EXECUTE) by entry at 9

CONTINUE RUNNING after JST by entry at 19 or press INT.3

To SET Q while at (WAIT) or (EXECUTE) enter at 11.

The computer will then wait for the next entry at CONT.

TRANSLATION or RUN. Alternatively, INT 1 will SET Q at any time during translation or running

During Translation

ERR 1	Inadmissible character - no decode. (May be due to a mis-punched tape. The tape stops when this error is found.)
2	Unacceptable character in this position. Character other than letter, digit + - , (or) found.
3	Number out of range. (Integer exceeding 131071)
4	Compiling program out of store range. Due to omission of (STORE) directive.
5	Error in directive letters.
6	(Integer not allowed in directive (Title string does not begin on newline
7	Modifier not Index Register 0 - 9
8	Error in Menonic letters
9	Location address, or literal, out of range during loading

During Running

ERR 10	Jump to location out of store range.
11	Program obeying number not instruction.
12	Contents of modifier not a number.
13	Location address out of range during running.
14	Card instruction.
15	Arithmetic or Output attempted with instruction not number.
16	Inadmissible character to output (Due to PCT using location not filled by RCT instruction)
17	RUN before EXECUTE directive e.g. if RUN is given at a (WAIT).
18	Floating point overflow (e.g. dividing by zero).