CESIL

Computer Education in Schools Instructional Language

Language and Compilers

Author: B W Spoor
D Bonham

Date: May 2023
Issue: 1 (Initial)

Contents

1. Introduction

CESIL was created by the ICL Education division (ICL-CES), later sold to Acorn Computers in 1983, as an introduction to computer programming in the late 1960s.

They produced a well structured and written series of books 'ICL-CES Computer Education in Schools, Computer Studies, Books 1, 2 & 3', texts about programming to work towards 'O' and 'A' levels.

There were a further two books 'Information Retrieval in Schools (IRIS)', which used a subset of the FIND-2 data interrogation and retrieval package, along with a few handy databases files for teaching databases and querying. As well as 'Symbolic Input Routine (SIR) Assembler Interpreter : Documentation and Listing'.

Computer Studies Book 1 starts with programming in CESIL, moving onto using BASIC, while Book 2 is BASIC only. The languages described in these books are implemented by the following 1900 programs:

- **#JCES**, a batch interpreter that runs under any of basic Executive, GEORGE 2, GEORGE 3, MINIMOP or MAXIMOP across most of the 1900 range and the 2903, processing batches of CESIL programs and data from either cards or paper tape.

- **# JBAS** an interactive interpreter that runs under GEORGE 3 or MAXIMOP.

When we started looking at this, #JCES appeared to be lost, but working copies of #JBAS exist and run quite happily using a 1900 emulator. A copy of the original compilation listing for #JCES has been located and the source rekeyed and compiled, restoring the original CESIL system.

While #JCES was thought to be lost, a new full compiler (#BCES) was written to re-implement CESIL.

For more information on CESIL, see https://iclces.uk/

2. CESIL Language

CESIL contains 14 instructions and a single accumulator, with a range of -8,388,608 to +8,388,607. The programs are generally written on coding sheets in four columns:

- Label (optional)
- Statement (mandatory)
- Operand (mandatory for most, but not all statements)
- Comment (optional and ignored apart from being listed)

At least a single space must separate these fields.

A label must be alphanumeric, starting with a letter and up to 6 characters long.

The statement must match one of CESIL's instructions, but only the first 3 characters are required.

The Operand, where required, may either be a variable name or a numeric constant (where permitted). Like a label, a variable name must be alphanumeric, starting with a letter and up to 6 characters long. A numeric constant takes the form of +nnn or -nnn and must be in the range of -8,388,608 to +8,388,607.

A PRINT string is a series of characters enclosed by the quotes character.

A comment is not processed and can be of up to the remaining character positions on the card or paper tape block.

A line starting with '(' is treated as comment.

CESIL has the following limits:

- No more than 50 distinct constants
- No more than 50 labels
- No more than 100 identifiers
- No more than 100 lines of CESIL
- No more than 400 items of data
- No more than 50 PRINT statements
- No more than 1000 characters of PRINT text

2.1 LOAD

The LOAD statement can have either a variable or constant as its operand and loads either the contents of the variable or constant value into the accumulator. It takes the format:

```
LOAD      COUNT
LOAD      +1234
LOAD      -64
```

2.2 STORE

The STORE statement can only have either a variable as its operand and stores the value in the accumulator in the specified variable. It takes the format:

```
STORE      COUNT
```

2.3 ADD

The ADD statement can have either a variable or constant as its operand and adds either the contents of the variable or constant value to the accumulator. It takes the format:

```
ADD       COUNT
ADD       +1234
```

2.4 SUBTRACT

The SUBTRACT statement can have either a variable or constant as its operand and subtracts either the contents of the variable or constant value from the accumulator. It takes the format:

```
SUBTRACT  COUNT
SUBTRACT  +1234
```

2.5 MULTIPLY

The MULTIPLY statement can have either a variable or constant as its operand and multiplies the contents of the accumulator by either the contents of the variable or constant value leaving the result in the accumulator. It takes the format:

```
MULTIPLY  COUNT
MULTIPLY  +1234
```

Note that MULTIPLY can produce a result outside of the range limits given above, which should be trapped as an error.

2.6 DIVIDE

The DIVIDE statement can have either a variable or constant as its operand and divides the contents of the accumulator by either the contents of the variable or constant value leaving the result in the accumulator. It takes the format:

```
DIVIDE    COUNT
DIVIDE    +1234
```

Note that DIVIDE rounds towards zero, so 8/3=2, -8/3= -2. Division by zero is trapped as an error.

2.7 JIZERO

The JIZERO statement will jump to the specified label if the contents of the accumulator have a value of zero. It takes the format:

```
JIZERO    AGAIN
```

2.8 JINEG

The JINEG statement will jump to the specified label if the contents or the accumulator have a negative value. It takes the format:

```
JINEG     AGAIN
```

2.9 JUMP

The JUMP statement will unconditionally jump to the specified label. It takes the format:

```
JUMP      AGAIN
```

2.10 IN

The IN statement causes the next data item to be loaded into the accumulator. It takes the format:

```
IN
```

2.11 OUT

The OUT statement causes the current contents of the accumulator to be converted to decimal and added to the print buffer. It takes the format:

```
OUT
```

2.12 PRINT

The PRINT statement causes the specified text string to be added to the print buffer. It takes the format:

```
PRINT     "text string"
```

2.13 LINE

The LINE statement causes the current line to be printed and the print buffer filled with spaces. It takes the format:

```
LINE
```

2.14 HALT

The HALT statement terminates the program run. It takes the format:

```
HALT
```

3. CESIL Program Layout

A CESIL program starts with a header consisting of 3 cards (or paper tape blocks), the first beginning with '**' identifies the school, the second beginning with '*C' identifies the pupil, the third is a program identification which follows the same rules as label or identifier.

The lines of code follow, terminated by a card (p/tape block) containing '%' in the first column. The data then follows, multiple items on a card are permissible, terminated with a '*' in the next data item position. The batch is terminated with a card or paper tape block having '****' in the first 4 columns.

```
**SAINT CUSTARDS
*CNIGEL MOLESWORTH
GCD
       PRINT    "         A         B      GCD"
       LINE
NXTGCD IN
       JINEG    NOMORE    NEGATIVE VALUE MARKS END OF DATA
       OUT
       STORE    A
       IN
       OUT
       STORE    B
AGAIN  JIZERO   DONE      IF B IS REDUCED TO 0, A IS THE GCD
       LOAD     A
       DIVIDE   B
       MULTIPLY B
       MULTIPLY -1        CHANGE SIGN SO ADDING TO A
       ADD      A         GIVES US A MOD B
       STORE    MODULO
       LOAD     B         SET NEW A TO OLD B
       STORE    A
       LOAD     MODULO    NEW B = A MOD B
       STORE    B
       JUMP     AGAIN
DONE   LOAD     A
       OUT
       LINE
       JUMP     NXTGCD
NOMORE HALT
%
8 12
9 7
54 24
42 56
-1
*
**SAINT CUSTARDS
*CBASIL FOTHERINGTON-TOMAS
NEWTON
( CALCULATE THE SQUARE ROOT USING NEWTON-RAPHESON
        IN
        STORE    X
        LOAD     +1
AGAIN   STORE    R
        LOAD     X
        DIVIDE   R
```

```
        ADD      R
        DIVIDE   +2
        STORE    NEWR
        OUT
        SUBTRACT R
        JIZERO   CLOSE
        LOAD     NEWR
        JUMP     AGAIN
CLOSE   HALT
%
169
*
****
```

4. Extended CESIL

When using the CESIL compiler #BCES, the language has been extended. There are two additional mathematical functions available and the ability to read and write data from/to a magnetic tape. Up to ten magnetic tapes may be open in either read or write mode simultaneously. This allows data to be created in one program and further processed in another, or intermediate results reprocessed within the same program.

The tape format conforms to ICL 1900 Housekeeping standards. The block size is 128 words with the first word of the block being a count of the number of valid data items (each a single word) in the block, i.e. up to 127 data items may be stored in a block. The number of blocks is limited by the size of the tape, continuation tapes are not permitted.

The additional commands are:-

4.1 MODULO

The MODULO statement can have either a variable or constant as its operand and divides the contents of the accumulator by either the contents of the variable or constant value leaving the remainder in the accumulator. It takes the format:

```
MODULO    COUNT
MODULO    +1234
```

4.2 NEGATE

The NEGATE statement causes the current value of the accumulator to be negated, i.e. if the initial value is +8 the result of the negate operation is -8, similarly -8 becomes +8. It takes the format:

```
NEGATE
```

4.3 OPEN

The OPEN statement causes the specified magnetic tape to be opened in either read or write mode. It takes the format:

```
OPEN      "unit,mode,filename"
```

Where unit is the MT unit number in the range 0 to 9, mode is either ‘R’ or ‘W’ and the filename is an alphanumeric text string of up to 12 characters, starting with a letter.

.

4.4 READ

The READ statement causes the next data item held on the specified magnetic tape to be loaded into the accumulator. It takes the format:

```
READ      unit
```

4.5 WRITE

The WRITE statement causes the current contents of the accumulator to be written to the specified magnetic tape. It takes the format:

```
WRITE     unit
```

4.6 CLOSE

The CLOSE statement causes the specified magnetic tape to be closed, any data in the buffer will first be written. It takes the format:

```
CLOSE     unit
```

5. Installing CESIL

The CESIL issue tape is in an extended library tape format that allows its use in both manual Executive and GEORGE 3 environments: containing the programs, subroutine library and G3 macros. Currently macros for G2 and MAXIMOP are unavailable.

5.1 Manual Executive

Normal use of #XPEU and #XPES will extract the programs and subroutines from the tape and install then in the required disc libraries. The disc files GCESSEMICOMP and SUBGROUPSCES need to be created, both with 1 block buckets and a minimum of 40 blocks allocated. #JCES and #BCES can be loaded into core directly from the issue tape if required, using the normal FIND command.

5.2 GEORGE 3 Environment

The program #JCES and #BCES along with SUBGROUPSCES should be installed into :LIB with the usual trap settings using #XPEU and #XPES. This may be done manually or automatically using macros from the central operators' console:-

```
RUNXPEU JCES,1,tsn
RUNXPEU BCES,1,tsn

RUNXPES SCES,1,tsn
```

The associated macros need to be copied into :MACROS, this can be achieved by running the following job (or issuing the commands in a MOP session).

```
JOB CESILFILES,:MACROS
FILEIN T????
,PROGRAM CESL
CESIL
CESILCOMP
CESILRUN
????
ENDJOB
****
```

5.3 Compiler Source

Additionally, the issue tape contains a further composite subfile 'SOURCE' containing the source subfiles for #BCES ('SOURCE BCES') and SUBGROUPSCES ('SOURCE SCES'), which may be extracted to cards using the magnetic tape editor #XKYA.

Currently the source for #JCES isn't being issued due to possible copyright issues.

6. #JCES – CESIL Interpreter

6.1 TITLE

To interpret and run a batch of CESIL programs on card or paper tape.

6.2 HARDWARE REQUIREMENTS

4032 words of core store
1 paper tape reader (optional)
1 card reader (optional)
1 line printer

6.3 USE OF PERIPHERALS

TR0	CESIL Source	The paper tape reader is optionally allocated at the start of run and released when the source has been read
CR0	CESIL Source	The card reader is optionally allocated at the start of run and released when the source has been read
LP0	Compilation Listing	The line printer is allocated at the start of run and released at end of run

6.4 DESCRIPTION

#JCES is the original ICL-CES CESIL interpreter, reading a batch of source programs/data from either cards or paper tape and running them.

The output is always sent to the line printer.

6.5 PROGRAM SWITCHES

Switch 0 ON – Disables the paper limit of 200 printed lines

Switch 1 ON – Enables monitoring

Switch 2 ON – If SW1 on, prints compiled version of CESIL program

Switch 3 ON – If SW1 on, prints JUMP table dictionary

Switch 4 ON – If SW1 on, prints store location identifiers dictionary

Switch 5 ON – If SW1 on, prints constants list

Switch 6 ON – If SW1 on, prints PRINT statement literals

Switch 7 ON – If SW1 on, prints data list

Switch 9 ON – Disables the time limit of 1000 JUMPs

6.6 OPERATING INSTRUCTIONS

1.	Load/Find #JCES	HALT:- LD
2.	Set any switches required	
3.	To start the program	
	with the source on paper tape	GO #JCES 20
	with the source on cards	GO #JCES 21
4.	At successful end of run	DLTD:- JJ

6.7 EXCEPTION CONDITIONS

Message	**Meaning/Action**	
HALT:- TR	Make paper tape reader available	GO to continue
HALT:- CR	Make card reader available	GO to continue
HALT:- LP	Make line printer available	GO to continue

7. #BCES – CESIL Compiler

7.1 TITLE

To compile a CESIL program to disc, ready for consolidation.

7.2 HARDWARE REQUIREMENTS

3200 words of core store
1 paper tape reader (optional)
1 card reader (optional)
1 card punch (optional)
1 line printer
1 disc cartridge

7.3 USE OF PERIPHERALS

TR0	CESIL Source	The paper tape reader is optionally allocated at the start of run and released when the source has been read
CR0	CESIL Source	The card reader is optionally allocated at the start of run and released when the source has been read
CP0	Semicompiled	The card punch is optionally allocated at the start of run and released at end of run
LP0	Compilation Listing	The line printer is allocated at the start of run and released at end of run
DA0	Semicompiled	The disc file GCESSEMICOMP is allocated at the start of run and closed at end of run

7.4 DESCRIPTION

#BCES is a full extended CESIL compiler, reading the source from either cards or paper tape and writing the semicompiled to disc, with automatic consolidation if the compilation is successful. The resulting binary is loaded into core, rather than to disc.

A listing is always sent to the line printer and a copy of the semicompiled may optionally be sent to the card punch for diagnostic purposes.

7.5 PROGRAM SWITCHES

Switch 0 ON – Send a copy of the semi compiled to the card punch

7.6 OPERATING INSTRUCTIONS

1.	Load/Find #BCES	HALT:- LD
2.	Set any switches required	
3.	To start the program	
	with the source on paper tape	GO #BCES 20
	with the source on cards	GO #BCES 21
4.	At successful end of run	DLTD:- FI#XPCK

7.7 EXCEPTION CONDITIONS

Message	**Meaning/Action**	
HALT:- TR	Make paper tape reader available	GO to continue
HALT:- CR	Make card reader available	GO to continue
HALT:- LP	Make line printer available	GO to continue
HALT:- CP	Make card punch available	GO to continue
HALT:- DA	Ensure the required disc file is available	GO to continue
HALT:- DR	Disc read error	Abandon run
HALT:- DW	Disc write error	Abandon run
DLTD:- CE	Compilation errors	Correct source and rerun

8. #GCES – CESIL Compiled Program

8.1 TITLE

Compiled CESIL program left in core, ready to run.

8.2 HARDWARE REQUIREMENTS

unknown words of core store
1 paper tape reader (optional)
1 card reader (optional)
1 line printer

8.3 USE OF PERIPHERALS

TR0	CESIL Source	The paper tape reader is optionally allocated at the start of run and released when the source has been read
CR0	CESIL Source	The card reader is optionally allocated at the start of run and released when the source has been read
LP0	Compilation Listing	The line printer is allocated at the start of run and released at end of run

8.4 DESCRIPTION

#GCES is a compiled CESIL program, loaded into core by the consolidation process. No copy is written to disc.

The program is re-entrant, in that it can be run multiple times with different sets of data from either the card reader or paper tape reader. Not all runs need to use the same input device. A listing is always sent to the line printer.

8.5 PROGRAM SWITCHES

Switch 0 ON – Disables the paper limit of 200 printed lines

8.6 OPERATING INSTRUCTIONS

1.	Automatically loaded by consolidator	HALT:- LD
2.	Set any switches required	
3.	To start the program	
	with the data on paper tape	GO #GCES 20
	with the data on cards	GO #GCES 21
4.	At successful end of run	HALT:- OK

8.7 EXCEPTION CONDITIONS

Message	**Meaning/Action**	
HALT:- TR	Make paper tape reader available	GO to continue
HALT:- CR	Make card reader available	GO to continue
HALT:- LP	Make line printer available	GO to continue
HALT:- ER	Errors in run	Correct data and rerun
HALT:- MT	Magnetic tape not available	Abandon run
HALT:- MR	Magnetic tape read error	Abandon run
HALT:- MS	Magnetic tape write error	Abandon run
HALT:- MW	Magnetic tape write error	Abandon run

9. SCES – Runtime Library for Compiled Programs

SUBGROUPSCES is the library file associated with the CESIL compiler #BCES. It contains the various subroutines used to implement certain CESIL instructions and basic control routines.

The subroutines are:-

%CESINIT	Initialise, contains entry points
%CESERR	Error handler
%CESTERM	Implements ‘HALT’
%CESLINE	Implements ‘LINE’
%CESMULT	Implements ‘MULTIPLY’
%CESDIV	Implements ‘DIVIDE’
%CESMOD	Implements ‘MODULO’
%CESIN	Implements ‘IN’
%CESOUT	Implements ‘OUT’
%CESPRNT	Implements ‘PRINT’
%CESOPEN	Implements ‘OPEN’
%CESREAD	Implements ‘READ’
%CESWRIT	Implements ‘WRITE’
%CESCLSE	Implements ‘CLOSE’
%CESCLSE	Null routine when MT routines not included

10. GEORGE 3 Macros

Three macros have been provided with the CESIL system. The first runs the original CESIL batch translator #JCES, the other two are from use with compiler #BCES.

10.1 CESIL

This macro will run a batch of CESIL source programs using the original translator #JCES. This macro may be used:-

```
CESIL parameters
```

Where the parameters are:-

```
*CRfilename – batch of source programs/data from cards
*TRfilename – batch of source programs/data from paper tape
NOLPLIMIT – sets switch 0 to allow more than 200 lines of output
NORUNLIMIT – sets switch 9 to allow more than 1000 jumps
MONITOR – sets switches 1..7 to invoke the monitoring facilities
```

Either *CR or *TR must be provided, the other parameters are optional.

For example:-

```
CESIL *CRFIRSTBATCH,NOLPLIMIT
```

10.2 CESILCOMP

This macro will compile a CESIL program using #BCES, saving the binary program for running using CESILRUN. This macro may be used:-

```
CESILCOMP parameters
```

Where the parameters are:-

```
*IDxxxx – program name (4 chars)
DEBUG – sends a copy of the semicompiled to cards
```

For example:-

```
CESILCOMP *IDGCES
```

In this case the source will be held in file ‘SOURCE GCES’, the compiled binary will be written to ‘PROGRAM GCES’ and the compilation listing in ‘GCESCOMPILST’.

10.3 CESILRUN

This macro will run a CESIL program compiled and saved by the macro CESILCOMP. This macro may be used:-

```
CESILRUN parameters
```

Where the parameters are:-

```
*IDxxxx – specifies 4 character program name
*CRfilename – specifies the data file (cards)
*TRfilename – specifies the data file (paper tape)
NOLPLIMIT – sets switch 0 to allow more than 200 lines of output
```

If the *ID parameter is not supplied, the program name will default to GCES.

If neither *CR or *TR are provided, the DATA-xxxx will be used for the data filename.

For example:-

```
CESILRUN *IDGCES,NOLPLIMIT
```

In this case the program source will be loaded from ‘PROGRAM GCES’ and the data read from ‘DATA-GCES’.

11. GEORGE 2 Macros

Two macros for GEORGE 2have been provided with the CESIL system. The first runs the original CESIL batch translator #JCES, the other uses the extended compiler #BCES. These macros can be extracted from the issue tape, using #XKYA (see TP4431: Compiling Systems), to either cards or a line printer listing. The subfile names being CESIL-G2 and ECESIL-G2.

The two macros can then be added to the standard G2 Macro Library using #XMED, as described in TP4432: GEORGE 2 Disc-based Operating System.

11.1 CESIL

This macro will run a batch of CESIL source programs held in DOCUMENT %A-BATCH using the original translator #JCES, the job being input from either cards or paper tape.

Parameter	***Function***	***Example***	***Default***
%A	Specifies the program name	JCES	JCES
%B	Specifies that switch 0 (no LP limit) is to be set	0	
%C	Specifies the output volume limit	10000	10000
%D	Specifies the library name	COMP	COMP
%E	Specifies that switch 9 (no jump limit) is to be set	9	

For example:-

```
CESIL
CESIL JCES,0,,COMP,9
```

11.2 ECESIL

This macro will compile and run an extended CESIL program held in DOCUMENT CESIL-SRCE using #BCES, the data being held in DOCUMENT CESIL-DATA, the job being input from either cards or paper tape.

Parameter	***Function***	***Example***	***Default***
%A	Specifies the program name	BCES	BCES
%B	Specifies that switch 0 (no LP limit) is to be set	0	
%C	Specifies the output volume limit	10000	10000
%D	Specifies the library name	COMP	COMP

For example:-

```
ECESIL
ECESIL ,0
ECESIL BCES,,,COMP
```

12. MAXIMOP Macros

Two macros for MAXIMOP have been provided with the CESIL system. The first runs the original CESIL batch translator #JCES, the other uses the extended compiler #BCES. These macros can be extracted from the issue tape, using #XKYA (see TP4431: Compiling Systems), to either cards or a line printer listing. The subfile names being CESL and ECSL.

The two macros can then be input to MAXIMOP's MACROS in the normal manner. It is recommended that the programs #JCES, #BCES & #XPCK are included in the library file PROGRAM MAXI, but it is not mandatory.

12.1 CESL

This macro will run a batch of CESIL source programs using the original translator #JCES.

Parameter	*Function*	*Example*	*Default*
%A	Specifies the batch source file	CTST	CONS
%B	Specifies the output destination	CONS	SOF
%C	Specifies that switch 0 (no output limit) is to be set	0	
%D	Specifies that switch 9 (no jump limit) is to be set	9	
%E	Specifies the library name	COMP	MAXI

For example:-

```
CESL CTST,,0,9
```

12.2 ECSL

This macro will compile and run an extended CESIL program using #BCES.

Parameter	*Function*	*Example*	*Default*
%A	Specifies the source file	BTST	CONS
%B	Specifies the data file	BDAT	CONS
%C	Specifies that switch 0 (no output limit) is to be set	0	
%D	Specifies the output destination – compiler	CONS	SOF
%E	Specifies the output destination – consolidator	CONS	SOF
%F	Specifies the output destination – CESIL program	CONS	SOF
%G	Specifies the library name (#BCES)	COMP	MAXI
%H	Specifies the library name (#XPCK)	COMP	MAXI

For example:-

```
ECSL BTST,BDAT,0
```