

## Thimbleby Tapes Investigation

Andrew Herbert

Updated: 4<sup>th</sup> December 2018

### Tapes Received

THIMBLEBY\_T1: unlabelled  
THIMBLEBY\_T2: 15/6/76? E/- mina  
THIMBLEBY\_T3: ML/1 IN (GPM) LOWL (3)  
THIMBLEBY\_T4: Saved ML1 source 17 Nov 76  
THIMBLEBY\_T5: unlabelled  
THIMBLEBY\_T6: Old version, enter at 16, run at 32  
THIMBLEBY\_T7: DISC PATCHER SOURCE 28/3/77  
THIMBLEBY\_T8: CO4 27.2.76 FIRST  
THIMBLEBY\_T9: Rados disc IO routines  
THIMBLEBY\_T10: CUR016 30/3/76 18.40 Algebra  
THIMBLEBY\_T11: GPM  
THIMBLEBY\_T12: 9/76 MLI TESTS

### THIMBLEBY\_T1

905 Disc Patcher MASIR source, dated "last mod 1/4/77". 900 telecode.

### THIMBLEBY\_T2

Assembly code possibly for a CTL Modular One? ASCII.

### THIMBLEBY\_T3

ML/1 version AIG, LOWL in GPM format. No matching macro definitions found, and some macros referenced are not in THIMBLEBY\_T11. 900 telecode. The missing tools have now been recreated.

This is presumably the pre-processed original LOWL as described in project report.

### THIMBLEBY\_T4

ML/1 in MASIR. Later version (3/11/76) than the one in Terry Froggatt archive (13/10/76). 905 disc file format. Appears to be same file as the earlier version with hand optimization of MASIR to remove redundant instructions. The MI-LOGIC of both appears to have been generated from the GPM format ML/I LOWL.

## THIMBLEBY\_T5

Binary version of L Game for 905. Sum-check binary for loading under initial instructions.

## THIMBLEBY\_T6

SIR source for L Game. 900 telecode.

## THIMBLEBY\_T7

905 Disc patcher MASIR source dated "last mod 28/3/1977" (see T1). 900 telecode.

## THIMBLEBY\_T8

LOWL translator test program in pre-processed format for GPM translator. 900 telecode.

## THIMBLEBY\_T9

ML/1 disc routines with TEST flag set to YES. 900 telecode.

## THIMBLEBY\_T10

Source of ALGEBRA package in LOWL. Same as 9/4/1974 ALGEBRA source held by Bob Eager but with slightly different tabulation. In original LOWL syntax, unpreprocessed. 900 telecode.

## THIMBLEBY\_T11 – GPM

900 telecode SIR source for GPM.

Described in report "K300 Computer Science Project 1976 – A General Purpose Macrogenerator and its applications – H.W. Thimbleby".

The program is a translation of Strachey's CPL. Additional/changed features:

1. BIN and DEC not required. BAR operations are +, -, \*, /, R (remainder). BAR takes text input and produces text output. Each BAR numerical argument is scanned left to right. Text preceding sign or first digit of number is ignored. Text following last digit is ignored. Sign characters are treated as monadic operators, thus -- is +.
2. New inbuilt macro LEG: [ LEG, n, L, E, G ] results in L, E or G being evaluated depending on whether the number is less than zero, zero, or greater than zero.
3. New macro NOTE: [ NOTE, arg ] outputs its argument as a monitor message to the teleprinter.
4. The document mentions further built-in macros, COND, TRACE and UNTRACE but these are not present in the code.

5. A character “!” followed by consecutive newlines evaluates to nothing.
6. It is possible to define a different set of warning characters.

Does not start by printing out program name and entry points as stated in document. If auto-triggered to label SB, listing of built-in macros fails as variable E is not set. Trigger and 4 SETE 5 E sequence added at SB.

First line BB=1 wrong, changed to B=1.

NEXTCH doesn't return correct value after reading non-warning character from tape. Redundant 1 -2 removed.

Jumps to LOAD+2 in START/COPY should be jumps to COPY to match Strachey's CPL – the attempted optimization fails on Strachey examples like FNPROD. The loop in LOADARG copying out the located parameter copied a bogus character if the parameter was of length zero. 7 START added before entry to loop.

An updated source TGPM has been made with the above changes. This has been tested on a significant number of sample GPM inputs, modified for the differences with HGPM and all execute satisfactorily. The small workspace (3,000 words) limits the more aggressively recursive ones to simpler cases).

## THIMBLEBY\_T12

ML/I Test suite. An older version than current set on Bob Eager's ML/I web site <http://www.ml1.org.uk>. 900 telecode. The ML/I in THIMBLEBY\_T4 passes these tests.

## Not found

1. Algol program to pre-process LOWL into form suitable for GPM translator. (This program is mentioned in Thimbleby's project report; it could be easily reverse engineered from inspection of the LOWL tapes).
2. A version of “Thimbleby GPM” with in-built COND macro (mentioned in project report), and also in-built macros required for LOWL translator (string decoding and identifier hashing). COND can be reconstructed from the specification in the report. The string decoding and hashing macros could easily be reverse engineered by comparing the ML/I LOWL and MASIR code).
3. LOWL translator source (i.e., macros corresponding to LOWL “instructions and directives”).

## RECONSTRUCTED SYSTEM

The port of ML/I for the Elliott 905 is based on a project by Harold Thimbleby in 1976 to port ML/I. The surviving tapes from this project were provided by Thimbleby and used as a starting point. The result of the project is a tool chain for the LOWL machine independent assembly code.

LOWL is an assembly code for an abstract machine designed by Peter Brown, initially at the University of Cambridge and subsequently the University of Kent. It originated as a means to allow his ML/1 macrogenerator to be ported to a wide range of different machines. Later Peter wrote other programs in LOWL – ALGEBRA, SCAN and REDUCE.

Details about LOWL can be found on the ML/I web site maintained by Bob Eager at <http://www.ml1.org.uk>. Some further documents missing from that archive are in the software documents folder of my archive.

From Thimbleby's tapes I have reconstructed the whole ML/I porting process, giving me a LOWL tool chain that has been further used to port the LOWL test suite and another application by Peter Brown called "ALGEBRA" in addition to reconstructing Thimbleby's port of ML/I.

The following notes describe the various program and data source files in two folders: LOWL for the general LOWL tool chain and MLI for the generation of MLI from the MASIR source from Thimbleby\_T4.BIN.

The LOWL folder includes the generation of MLI from the tool chain, where as the version in the MLI folder is built from Thimbleby's MASIR source which is itself the result of further hand optimization applied to that generated by the tool chain.

Thimbleby ported LOWL using the following tools written as part of his project:

1. An Algol program to read the ML/I machine independent part ("MI Logic") LOWL as distributed by Peter Brown and convert it such that each LOWL instruction and direction mapped to a call of a GPM macro.
2. GPM macros to generate equivalent 905 MASIR assembly code to each LOWL instruction and directive.
3. A modified version of GPM including some additional support macros specifically for LOWL translation.
4. A set of library routines for the Elliott 905 Rados operating system to provide the LOWL machine dependent system interface ("MD Logic") for ML/I.

This tool chain has been recreated and used to derive ML/I from the original LOWL of ML/I version AIG. The tool chain has been verified by successfully running the LOWL test suite.

The main files are in the folder MASTERS/LOWL

ALGEBRA\_MDLOGIC.900: machine dependent logic for ALGEBRA application (derived from MLI\_MDLOGIC.900).

ALGEBRA\_MILOGIC.900: Machine independent logic for ALGEBRA application (from Bob Eager).

The LOWL has been modified to put a prompt output before reading each line of input, using the MESS instruction.

ALGEBRA\_REFS.900: Variable references for ALGEBRA built by hand.

EXEC.900: This file contains a minimal operating system based on SSYS1 to support LOWL\_SYSTEM. Note EXEC.900 treats halt code as an illegal character, since LOWL\_SYSTEM uses ')' on a line of its own to indicate end-of-file.

EXEC.900 is basically SSYS1 with some small changes:

1. CHIP returns -1 for an illegal character (including halt code).
2. Added definition of macro MQCHOP (Elliott manual says this is built-in, by my MASIR tape says it is "undefined").
3. Sets FFPTX and LFPTX labels
4. Code to drop to level 4 so that locations 8180-8191 are enabled as usable store locations.
5. Entry to MD Logic at the label START.

LOWL2GPM.900: SIR program to read LOWL input and convert to GPM format for subsequent expansion via LOWL\_TRANSLATOR macros.

LOWL\_GPM.900: This is the source of a modified version of Thimbleby's implementation of GPM (in MASTERS/THIMBLEBY). It has the following additional macros built-in:

ALPHA – ALPHA takes a LOWL string as its argument, e.g., 'THIS IS A STRING' and translates it to a sequence of MASIR alpha numeric constants, e.g., \THI \S I \S A \STR \ING.

The character \$ is translated to newline.

The macro will process multiple arguments so that a call such as :ALPHA,AB,CD,EF; will produce \AB, \CD, \EF. (Provided the LOWL\_GPM argument separator remains as comma.

COND – This macro provides a pattern matching capability.  
:COND,arg,val 1,res 1,...,val n,res n,default; results in the output of res i where val i is the first value to be identical to arg, otherwise default is generated.

STR – This macro works like ALPHA but outputs the characters in the string as octal constants representing the internal code assumed by the

LOWL system interface. Digits stand for themselves. Punctuation characters are of the form &5000nn where nn is the SIR internal code value for the character, and letters of the form &6000nn similarly. (Note: symbols outside of the SIR internal code alphabet give rise to a QCHIN error message). Constants of the form &54000n are used for various internal markers.

UNQUOTE - This macro expects a single argument in the form of a LOWL string and simply removes the surrounding quote characters ( ' ). Note: UNQUOTE produces an error for strings containing commas.

In addition, LOWL\_GPM recognizes LOWL string quotes ( ' . . ' ) and treats any LOWL\_GPM warning characters in the string as normal characters, overriding their GPM meaning.

LOWL\_TESTS\_MDLOGIC.900: The machine dependent logic of the LOWL test suite, derived from MLI\_MDLOGIC.900.

LOWL\_TESTS\_MILOGIC.900: the machine independent Logic of the LOWL test suite from Bob Eager.

LOWL\_TESTS\_REFS.900: list of address reference variables for LOWL\_TESTS\_MILOGIC.900 built by hand.

LOWL\_TRANSLATOR.900: This file contains GPM macros to translate LOWL code translated to GPM macro calls into MASIR. It requires the LOWL\_GPM version of GPM for the built-in ALPHA, COND, STR and UNQUOTE macros and the handling of LOWL strings.

MLIHASH.900: A program to generate the hash table of built-in identifiers. Input consists of a sequence of identifiers; output consists of definitions of HASH and THASH macros for use with LOWL\_GPM.

MLI\_MDLOGIC.900: This file contains the LOWL MD-LOGIC for ML/I (taken from Thimbleby's ML/I with modifications to remove Rados specific code and integrate with EXEC.900).

MLI\_MILOGIC.900: The LOWL for the MI Logic of version AJB of ML/I (from Bob Eager).

MLI\_REFS.900: A list of variable references required by the MD logic. Built by hand.

In the folder SCRIPTS/LOWL

BUILD\_ALGEBRA.DAT: Generates and assembles MASIR source of ALGEBRA to ALGEBRA.BIN

BUILD\_LOWL2GPM.DAT: Builds LOWL2GPM.BIN.

BUILD\_LOWL\_GPM: Builds LOWL\_GPM.BIN.

BUILD\_LOWL\_TESTS.DAT: Generates and assembles MASIR source of LOWL\_TESTS to LOWL\_TESTS.BIN.

BUILD\_MLI.DAT: Generates and assembles MASIR source of ML/I to MLI.BIN

In the folder MASTERS/MLI:

DOCUMENT.900: Thimbleby's notes on his implementation.

MLI\_EDIT.900: Bowdler steering tape to modify the source in THIMBLEBY\_T4.BIN to work with EXEC.900.

THIMBLEBY\_T4.BIN: Thimbleby's November 1976 distribution of ML/I.

In the folder SCRIPTS/MLI

BUILD\_MLI.DAT: script to unpack THIMBELBY\_T4.BIN and assemble ML/I.