# Abstracts of the 1988 Australian Forth Conference

## Forth in Computer Systems Engineering Education

*C.A. Maynard*
*Head, Department of Computer Engineering*
*Curtin University of Technology, Bentley, W.A. 6102*
*Wave-onic Associates*
*PO Box 607, Cannington, W.A. 6107*

This paper discusses the areas of application and the reasons for the choice of the Forth language in the training of Computer Systems Engineering students at the Curtin University of Technology.

## Teaching Forth

*Feisal Ramadan*
*School of Electrical Engineering, North Syndey College of TAFE*
*GORE-HILL, N.S.W. 2065, Australia*

Software Engineering Forth is the name of a post certificate course offered at North Sydney College of TAFE (NSCT) School of Electrical Engineering. Participants to the Forth Course come from the electronic industry and have as a minimum prerequisite a certificate in Electrical and Electronic engineering or equivalent qualification. The Forth course is free but the cost of books, software and hardware is borne by the participant. Attendance is 3 hours per week for 17 weeks. In one semester students learned a new language and assembled a Forth computer kit. They generated a turnkey application, metacompiled their own Forth program, wrote short machine language routines and gained insight in the working of an operating system and a modern language. Forth is an efficient and potent education tool. Our School has officially adopted the subject which can now be offered anywhere in the State.

(synopsis by L. Forsley)

## Computer Assisted Laboratories with ASYST

*Dr. A. Pubatshew*
*School of Applied Physics, South Australian Institute of Technology*

The intelligent and productive use of micro-computers in undergraduate Physics laboratories is an important opportunity to introduce a modern scientific tool and the associated techniques of data acquisition and analysis. The computer can be used in many ways — equipment control, data analysis and assisting the student comprehension of physics principles. The successful integration of these multi-purpose experiments requires a new view of the laboratory aims and its importance. In fact, effort must be made not to produce a completely automatic experiment just because there is a need for some computer based practicals. In this paper we discuss the results of a program commenced in 1985 that has the following objectives:

- Expand the experiments into the second and third year laboratories.
- Ensure productivity and efficiency with appropriate choice of language.
- Choose and design appropriate equipment.
- Design experiments that can be shared with other tertiary institutions.
- Design practicals that permit some degree of student participation in both design and programming.

## Simulations for CAL in Forth

*I.P. Walsh*
Scientific Officer
*D.E. Bailey*
Senior Lecturer
*School of Science and Technlogy, Nepean College of Advanced Education*

The Physics Literacy Project was initiated in 1984 with funding supplied by the Commonwealth Tertiary Education Commission under the Participation in Equity Program. The project had as its aim the production of computer assisted learning programs to assist students with little or no background in Physics to gain necessary concepts for success in tertiary Physics courses. The project was initiated using Apple Macintosh computers and the programming language Forth. Four years down the track, a number of packages have been generated that emphasize screen production of simulations of Physics events. The emphasis of this paper is not in the origins of the project and the final product obtained, nor of the uses to which it will be put, but on the lessons learnt in the process of producing the graphics sequences and on the language used, MacForth. The paper is written in the first person because it mainly concerns Ian Walsh's experiences as the most recent programmer with this project.

## Forth and Programmable Controllers

*Gary Brown*
Systems Engineer
*Australian Industrial Electronics Pty. Ltd.*
*47 Gatwick Road North, Bayswater, Victoria 3153*

AIE is the designer and manufacturer of the RTI500 programmable controller. All the software layers which make up the unit have been entirely implemented in Forth.

## Design Philosophy of an NC4016-based Microcomputer

*Roy Hill*

A very small (3 person) Australian electronics organization [was] able to design, produce and market an IBM type co-processor board that even the Americans would be proud to claim as their own. The board is designed to occupy one of the 8 bit slots in an IBM PC. However, the provision of a serial port that handles a standard RS232 interface means that it can also be used in an external mode. The addition of the A/D-D/A converter and associated support circuitry on-board also means that it is possible to develop applications for data acquisition/control, robotics... and high speed video capture.

(synopsis by L. Forsley)

## Protocol Testing Using Forth

*S.A. Leask*
*R.A. McNaughton*
*Telecom Research Laboratories, Clayton, Victoria*

A communications protocol is a set of rules which governs the communication between two or more entities such as computers, telephone exchanges, etc. The more complex protocols require sophisticated tools to verify their correct operation and analyze their performance. In December 1986 Telecom Research Laboratories and Siemens AG (Germany) embarked on a joint project to develop a test system for the Integrated Services Digital Network (ISDN) Access protocol. Currently, a second project is underway to develop a test system for the Common Channel Signalling number 7 (CCS7) protocol, which is used for the signalling between telephone exchanges. Both these systems are based on the Siemens K1195 Protocol Tester using the FORTH programming environment.

(synopsis by L. Forsley)

## Forth and Parallel Processing

*Paul Wilson*
*QIT*

In a series of experiments, Forths running on an MC 6809 based computer have been accepting data from computations executing on [Inmos] Transputers, and driving hardware interface circuits. This arrangement is part of a simulation computer, which solves differential equations at high speed and provides output data to analog recording instruments. The Forth computer has also been used for commissioning and troubleshooting Transputers, [and] has been used for 'capturing' data passed between two Transputers.

(synopsis by L. Forsley)

## ASYST Applications

*Michael G. Smart*

*Science and Computing Applications Party. Ltd*

ASYST is a FORTH-based language especiallly suited to engineering and scientific applications. In this paper I will discuss the primary productivity-related features of ASYST (many of them arising from its relationship to FORTH architecture). To illustrate the various features, I will refer to a recently completed Test Management and Data Acquisition System implemented in ASYST for Traffic Authority of New South Wales.

## PostScript for the Forth Programmer

*Suzanne W. Hogg*

*University of Technology, Sydney*

PostScript "programming" can be indulged in at many levels, from the complete control (but no WYSIWYG) of a fully designed layout written in quite a substantial amount of Forth-like code through to the complete WYSIWYG facilities with the amount of control allowed to the user by the ever-increasing in versatility software packages which translate directly into PostScript instructions. Artists, architects, engineers, scientists should find the coordinate specifications of the language easy to handle, while the desktop publisher who wants to add a special "something" to the layout should find it rewarding to study the PostScript language at least to the "Cookbook" level. The ability to design one's own fonts is like the "icing" on the cake.

## THInC — An Interactive and Extensible FORTH Compiler

*Klaus D. Veil*

*Director of Biomedical Research Development*
*PACE Corporation, Sydney, Australia*

The FORTH programming approach has been acclaimed as an effective productivity tool mainly due to its interactiveness and extensibility. However, the advent of native-code compilers for the other languages has resulted in quite unfavourable FORTH runtime benchmark comparisons. The Threaded High-level Interactive nano-Compiler (THInC) system has been implmented as an attempt to alleviate this shortcoming while maintaining all the productive features of FORTH.

## Using Execution Arrays & Time Based Operations in Forth

*D.C. Edwards*

*Director of Engineering*
*JARRAH Computers Pty. Ltd.*

Jarrah Computers has specialized in the development of custom designed industrial micro controllers — from the 68705 family of single chip controllers to education systems and large industrial control systems based on the Rockwell 65F11, the 6502 implementation of a single chip FORTH. FORTH has been used as the preferred development language for all of these applications and has provided Jarrah Computers with a consistent software environment for the wide range of development projects which it has undertaken. In this paper I shall discuss some specific uses to which Jarrah Computers has put FORTH, namely the syntaxes surrounding the use of execution arrays and our development of a system for the coding of time based events.

(synopsis by L. Forsley)

## Forth Engines vs Assembly (Forth Applications in industrial control)

*Ray Gardiner*

*Ardmona Fruit Products Co-op Co. Ltd.*
*PO Box 196 Mooroopna 3629, Victoria, Australia*

Programming real time control systems in Forth allows the interactive development and interactive debugging of hardware interfaces. The paper explores the issues of development techniques, software maintenance, execution time speed and complexity.

### Applications of Asyst in Digital Signal Processing

*R. Prandolini*
*M.P. Moody*
*J. Miorandi*
*School of Electrical and Electronic Systems Engineering*
*Queensland Institute of Technology*
*Brisbane, Australia*

ASYST is a FORTH based software package which consists of a Mathematics and Graphics Module, a Data Acquisition Module, a Data Analysis Module and a General Purpose Interface Bus (GPIB) Module. The wide range of inherent functions and the flexibility of programming allow sophisticated turn-key signal processing algorithms to be produced in relatively short time.

### Forth and Prolog on the Forth Machines

*L.L. Odette*
*Applied Expert Systems, Inc.*
*One Cambridge Center, Cambridge, Massachusetts 02142 USA*
*W. Wilkinson*
*School of Electrical Engineering*
*1269 Commonwealth Ave, #11, Boston, Massachusetts 02134 USA*

Our research focus is the problem of engineering expert system technology into systems with severe resource bounds, difficult performance constraints, and non-standard hardware. Good examples are embedded systems for intelligent real-time data acquisition and control. We argue that Forth can profitably be viewed as a platform language for both the real-time and knowledge-base portions of such systems. This paper describes our development work on a Prolog compiler that emits code for an abstract Prolog machine. The Prolog machine is implemented in Forth; versions have been built for the Novix Forth engine and a version is in development for the Harris RISC-based real-time control processor.

### ASYST Programming for Biomedical Engineering Applications

*Nigel Lovell*
*Branko Celler*
*Biomedical Engineering Laboratory, Department of System and Control*
*School of Electrical Engineering and Computer Science*
*University of NSW, Kensignton 2033*

The ASYST language is a Forth based language designed to provide a full range of mathematical, statistical and interface functions for the scientific user. Integral to the language are functions for computer interfacing via serial (RS232/422) devices, functions for interfacing using the IEEE-488 (GPIB) bus, and drivers for interfacing using plug in data acquisition cards.

Features of the ASYST language are described with reference to biomedical engineering application areas. Particular reference will be drawn to software developed in our laboratory for acquisition and analysis of ECG data. We have named this suite of ASYST programs CARDIOSYS. CARDIOSYS was the first application program to be written in the ASYST language in Australia.

The advantages of using ASYST over conventional languages will also be discussed.