RRE

CORAL 66

A NATIONAL STANDARD FOR REAL TIME

COMPUTING STANDARDS SECTION,
ROYAL RADAR ESTABLISHMENT,
PROCUREMENT EXECUTIVE,
MINISTRY OF DEFENCE,
MALVERN, WORCESTERSHIRE, ENGLAND

CORAL 66 A NATIONAL STANDARD FOR REAL TIME

CORAL, A Computer On-line Real time Applications Language, was originally devised by I F Currie and M Griffiths in 1966. The "Official Definition of CORAL 66", edited by P M Woodward, P R Wetherall, and B Gorman was first published by Her Majesty's Stationery Office in 1970 and a second impression was published in 1973. CORAL 66 was chosen by the Ministry of Defence as an interservice standard for military programming,and following its widespread use in military and now in civilian projects, it has been adopted as a standard for real time and process control systems by the Department of Trade and Industry, the Control and Automation Manufacturers Association, the Software Houses Association and the National Computing Centre.

INTRODUCTION

CORAL 66 is not intended to compete with established languages in use on large scientific or business machines. It is aimed mainly at implementers of systems with small dedicated computers where hitherto the use of high-level languages has by no means been universal. For a high level language to replace machine code in this type of system the compiler must be small enough to run in the production system or its standby system. It must produce efficient code comparable in size to that which would have been produced by an assembler. The Language must allow full use of individual machines hardware and any other special facilities, but at the same time be capable of implementation on a wide range of machines. Finally the Language must be cheap and quick to implement on new machines, and this process should be in part capable of being implemented automatically using such techniques as automatic syntax analyser generators. This paper then reviews CORAL 66 with these concepts in mind.

REAL TIME FACILITIES

Timing and interrupt facilities are not standardized in CORAL 66, as the language is intended to be suitable for a wide variety of computers with different supervisory software. The programmer's control over external events, and the computer's reaction to them, must be expressed by calls of procedures or macros with bodies designed to interface with whatever facilities are normally provided by the computer manufacturer. No fixed conventions are laid down, but the parameter mechanism for procedures and macros is sufficiently powerful to permit definition of useful real-time statements at the language level.

PROGRAM STRUCTURE

A program consists of communicators and separately compiled segments. Each segment has the form of an Algol 60 block, within which blocks and procedures may be nested to arbitrary depth. In the absence of communicators, block structure would prevent different segments from using common data, labels, switches or procedures. The purpose of a communicator is to specify and name those objects which are to be commonly accessible to all segments. One type of communicator, common, is used for all objects which are declared within the program, either in the common communicator itself (data), or in the outermost block of any segment (labels and procedures). Other forms of communicator are permitted for specifying objects whose full definitions are external to the program, eg library procedures, or for reference to objects having absolute addresses in the computer. The presence of communicators also imposes a modular and disciplined approach to programming larger systems where a team of programmers are employed.

DATA STRUCTURE

To attain maximum speed at run-time, static storage allocation is assumed. Array bounds must be constants. In addition to arrays with whole-word elements (one or two dimensions only), a "table" feature enables one-dimensional arrays to be built up from items which may extend over several words each. Particular fields within such an entry, with lengths of any number of bits, can be selected by means of field identifiers declared along with the table. A data overlay facility is provided, so that a given area of store can, if necessary, be simultaneously referenced in different ways.

DATA TYPES

Whole-word items are numbers of signed floating, fixed or integer types. Fixed-point numbers are declared with a specification of scale and significance, eg FIXED (13, 5) means 13-bit significant accuracy with 5 bits after the point. Short items in tables are declared as signed or unsigned fixed-point numbers. Short items may also be extracted from any whole-word or data, eg BITS [8, 5] x is an unsigned integer formed from bits 5 to 12 (8 consecutive bits) of x. Using the operator LOCATION, the programmer is able to obtain the machine addresses of data; the inverse operation is also provided.

ARITHMETIC AND LOGIC

Arithmetic expressions involving addition, subtraction, multiplication and division can be written between any of the data types, and between bracketed expressions and functions, which may be nested to any depth. Type changing and scaling are automatic, and rounding is introduced where necessary. The scaling of sub-expressions is normally undefined, but may be specified if required. Subscripts can be any expression, and are automatically evaluated to an integral value.

In addition to the logical operations such as AND and OR which can be used in conditions, parallel logic on whole words is included in the language. The operators are MASK (logical "and"), UNION (inclusive "or") and DIFFER (" not equivalent").

STATEMENTS

The statements of CORAL 66 are similar in form to those in Algol 60, with a few simplifications (no multiple assignment, no designational expressions), and with the addition of a "code statement" corresponding to the undefined code of the machine. Code instructions can use any CORAL 66 identifiers in scope at the time; indeed, the language definition lays down that this must be possible.

PROCEDURES

As in Algol 60, procedures may have values or not. The chief difference is that the Algol 60 "name parameter" has been replaced by a "location parameter", in which an address is calculated once only on entry to the procedure. A value parameter may be specified to any arithmetic type, and the actual expression is automatically evaluated to suit. Actual parameters of formal location parameters, however, must be of the correct type and scale. Arrays, tables, procedures, labels and switches are all permitted forms of parameter. For reasons of efficiency, procedures which are recursive must be explicitly declared as such; otherwise procedures may be declared and used with full generality subject to the overriding requirement that identifiers other than labels, must be specified or declared before they are used.

IMPLEMENTATION

CORAL 66 is designed so that the time taken to compile, load and execute is as short as is consistent with running the compiler in a moderate amount of core store with no backing store. Not more than one input and output paper tape channel need be used. Conceptually, all compilers can be single pass, though some in practice may be 2-pass so as to reduce the storage required. Typically, on a machine with a 4 microsecond add time, the time to compile 3000 instructions of object code will be less than 3 minutes in a system entirely dependent upon paper tape. A CORAL 66 compiler has been produced to run in a 4 K machine using 3 passes.

The insistence that every identifier (except for labels local to a block) is fully declared or specified before it is used simplifies the compiler by ensuring that all relevant information is available when required. Features which require elaborate hardware in the object machine for efficient execution of a program (eg dynamic storage) are not included in the language. The loader (consolidator, linkage editor) is not required to perform identifier matching between different segments, and is thus considerably faster than most commercial loaders.

In contrast to Fortran, no run-time checks are required on procedure entries, and there is no run-time storage housekeeping overhead such as Algol 60 generates. The block structure makes more efficient use of storage space than Fortran. The static algorithm used for storage allocation is not as economic as dynamic storage; it is used because it does not impose any burden of extra code at run-time, which is especially important on object machines which lack proper facilities for accessing dynamic storage. The compiler introduces no checking on subscripts and their bounds, unless a special diagnostic mode of compilation is used. Conditions are evaluated only as far as is necessary to determine their truth and falsity (in contrast to Algol 60). Particular compilers can introduce many other optimisations without requiring more than one basic pass.

USE

The applications that CORAL 66 is being used in are many and varied. Examples include the control of power stations, steel mills, radar antennas, air traffic control systems, message switches and computer operating systems. It is also extensively used in systems supporting software and as a language for the writing of new CORAL compilers. At the Royal Radar Establishment (RRE), the Computer Applications Division have two computer systems with the complete operating system, compiler, and all supporting software written in CORAL 66. It has been adopted by at least three major British computer companies for all their programming.

As a result of the wider adoption of the language as a standard, work has now been started to provide greater supporting facilities to CORAL 66 users. This includes the provision of training facilities, the setting up of a CORAL 66 library, and the establishment of User groups. The Computing Standards Section at RRE acts as a focal point to provide advice and assistance to users or potential users.

It is hoped that the provision of these support facilities will encourage the increasing use of the language both in Britain and Europe with consequent reductions in costs and the time taken to implement real time systems. Figures published by the Government's Central Computer Agency in their TSD note 4103 indicate that programmer productivity can increase from 4 to 15 times, with the consequent code being on average only 30% more.

For further information on CORAL 66 contact the Computer Applications Division, the Royal Radar Establishment, MALVERN, Worcs, ENGLAND. The telephone number is MALVERN (06845) 2733 Extensions 3063, or 2320, or 2165, or 2925.

RRE Malvern
(Revised Feb 1974)