T9000 occam 2 TOOLSET

IMS D4395, D7395

T9000 software development tools

PRODUCT INFORMATION

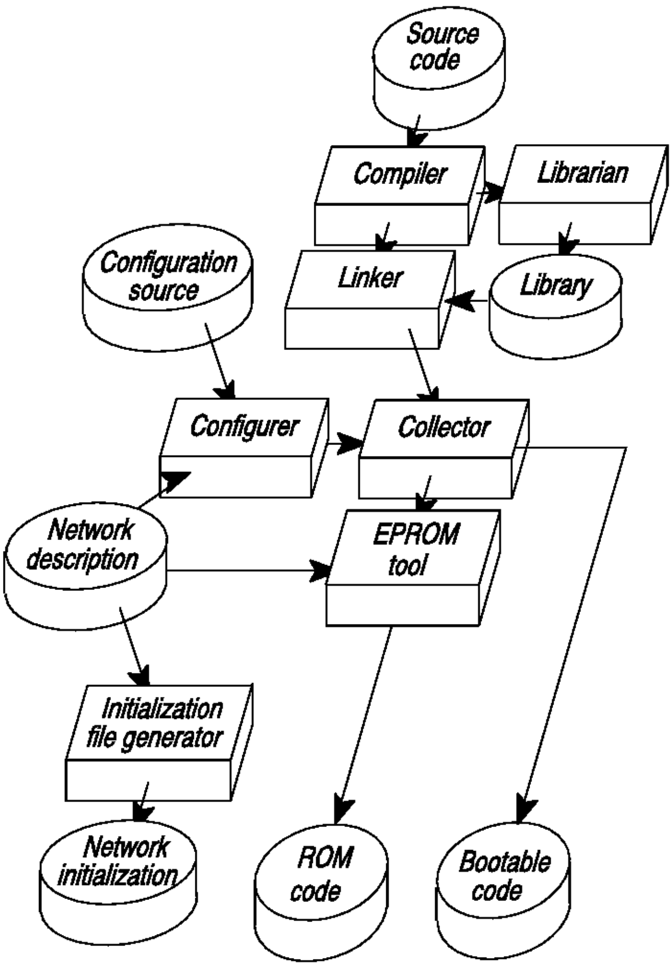
FEATURES

- Complete occam 2 toolset for IMS T9000 transputer networks.
- Source compatible with T2/T4/T8 and ST20 development tools.
- Cross-development from Sun 4 and 386 PC platforms.
- Code generation for IMS T9000 instruction set and pipelined CPU.
- Configuration tools that exploit new communications architecture.
- Software routing used for networks without routing chips.
- Support for mixing ANSI C and occam 2.
- Support for communication between T2/T4/T8 networks and IMS T9000 networks.
- Support for assembler inserts.
- Support for EPROM programming.

DESCRIPTION

The SGS-THOMSON T9000 occam 2 Toolset supports the construction of occam 2 programs which may be loaded onto IMS T9000 networks via a link, or put into a form suitable for booting from ROM.

The Toolset also includes the hardware configuration tools required to initialize IMS T9000 networks, fully supporting the use of STC104 packet routing switches including initialization and labelling of networks.



The information in this datasheet is subject to change 42 1589 03

Contents

1 Introduction

This document contains product information for the T9000 occam 2 Toolset. Toolsets are available for IMS T2xx, T4xx and T8xx series transputers. Within this document the IMS T2xx, T4xx and T8xx transputers are called 'T2/T4/T8' transputers.

The T9000 occam 2 Toolset provides a complete occam 2 cross-development system for IMS T9000 transputer network targets. It can be used to build sequential or parallel programs for single IMS T9000 transputers or for multi-transputer IMS T9000 networks.

The Toolset supports the construction of occam 2 programs, which may be loaded onto IMS T9000 networks via a link or put into a form suitable for booting from ROM. The Toolset also includes the hardware configuration tools required to initialize IMS T9000 networks. It fully supports the use of STC104 packet routing switches including labelling and initializing.

1.1 Key features

- occam 2 cross-compiler for IMS T9000 target networks
- Code generation for IMS T9000 instruction set and pipelined CPU;
- Scientific libraries
- Excellent compile time diagnostics;
- Support for parallelism;
- Support for assembler inserts;
- Listings of where variables and functions reside in memory;
- Small runtime overhead;
- Separate compilation, using linker and librarian tools;
- Tools for creating and loading multi-processor programs;
- Exploitation of the new communications architecture by the configuration tools;
- Software routing of channels for networks without routing chips;
- Automatic makefile generator;
- Mixed language programming support;
- Tools to support preparation of programs for EPROM;
- Consistent tools across PC and Sun-4 hosts;
- Support for dynamically loading programs and functions.

1.2 Overview

The IMS T9000 transputer range of devices is supported by the following software products:

- IMS Dx394 T9000 ANSI C Toolset;
- IMS Dx395 T9000 occam 2 Toolset;

- IMS Dx390 T9000 INQUEST development environment, containing debugging and profiling tools;
- IMS Dx397 interface software.

The T9000 occam 2 Toolset is source compatible with T2/T4/T8 and ST20 Toolset products. This has been achieved by using similar compiler, libraries, linker, configurer and other tools. In addition, code written for a single IMS T4xx or IMS T8xx can be ported directly to an IMS T9000 for performance enhancement.

The IMS T9000 offers two important enhancements; the new communications architecture and the new pipelined CPU. In order to take full advantage of these features, a new Toolset has been developed which will function alongside the existing Dx395 T9000 occam 2 Toolset.

The configuration tools are the main area in which the difference between the IMS T9000 and the T2/T4/T8 range is apparent to the programmer. Configuration is the process of mapping a multi-process network application to a transputer network. Tools are provided in the T9000 occam 2 Toolset allowing users to:

- describe a hardware network made up of IMS T9000 and STC104 packet routing switch devices;
- define values for the user-programmable attributes of the devices (such as the IMS T9000's programmable memory interface, and the STC104 packet routing switch's interval labelling registers);
- define how processes in the application should be mapped to processors in the network.

The tools allow the full user-programmable functionality of the IMS T9000 and STC104 devices to be used with simple textual descriptions of attribute values, without the need to write any low-level configuration code.

The configuration tools can check the network description to ensure that the network is properly connected. Applications of arbitrary connectivity can be mapped onto the hardware network. Where the network includes STC104 packet routing switches, the headers required to connect the channels between processors are calculated automatically by the tools. If the network does not contain STC104 packet routing switches, the tools provide software through-routing so that channels may still be connected between processors which are not neighbors.

EPROM tools are provided to support the creation of initialization ROMs for IMS T9000 transputers, and the creation of system ROMs containing application code.

The debugger supplied in the INQUEST development environment supports source-level and low-level debugging of multi-process and multi-processor programs. The debugger runs on the host computer from which the IMS T9000 network has been loaded. A graphical user interface, with multiple windows, simplifies use of this sophisticated tool. INQUEST also includes profiling tools for detecting program 'hot-spots' and for evaluating how effectively parallel programs use a network of processors.

1.3 Hosts

Programs developed using the Toolset are both source and binary compatible across all host development machines. The T9000 occam 2 Toolset is available for the following development platforms:

IMS D4395 T9000 occam 2 Toolset for Sun-4 under SunOS and Solaris, IMS D7395 T9000 occam 2 Toolset for IBM PC under MS-DOS.

2 IMS T9000 networks

The new generation transputer, the IMS T9000, provides major enhancements to the communications capabilities of transputer networks. The T9000 occam 2 Toolset provides support for these features in the form of new configuration and initialization tools.

Using the IMS T9000 transputer and the T9000 occam 2 Toolset, a large network of processes and channels may be programmed quite independently of the transputer network on which it will be implemented. The IMS T9000 can multiplex a large number of channels, called virtual channels, onto a single physical link. This allows a large network of channels between processes to be efficiently implemented on small IMS T9000 networks without routing or multiplexing software.

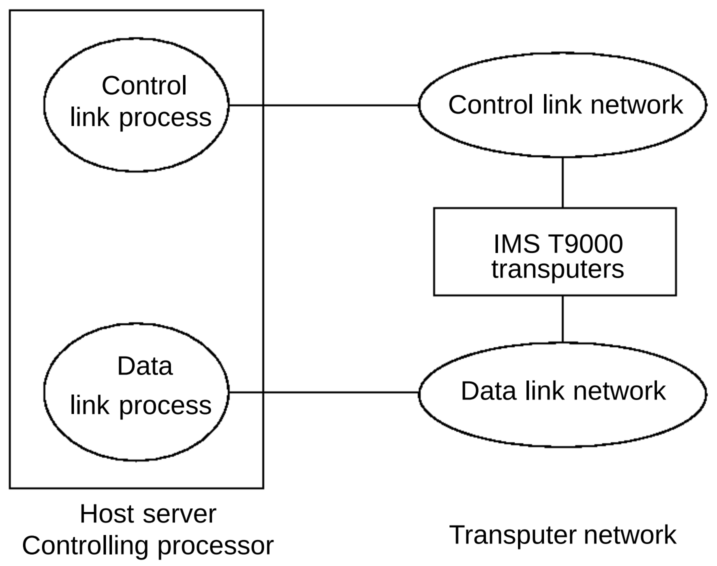


Figure 1 IMS T9000 network architecture

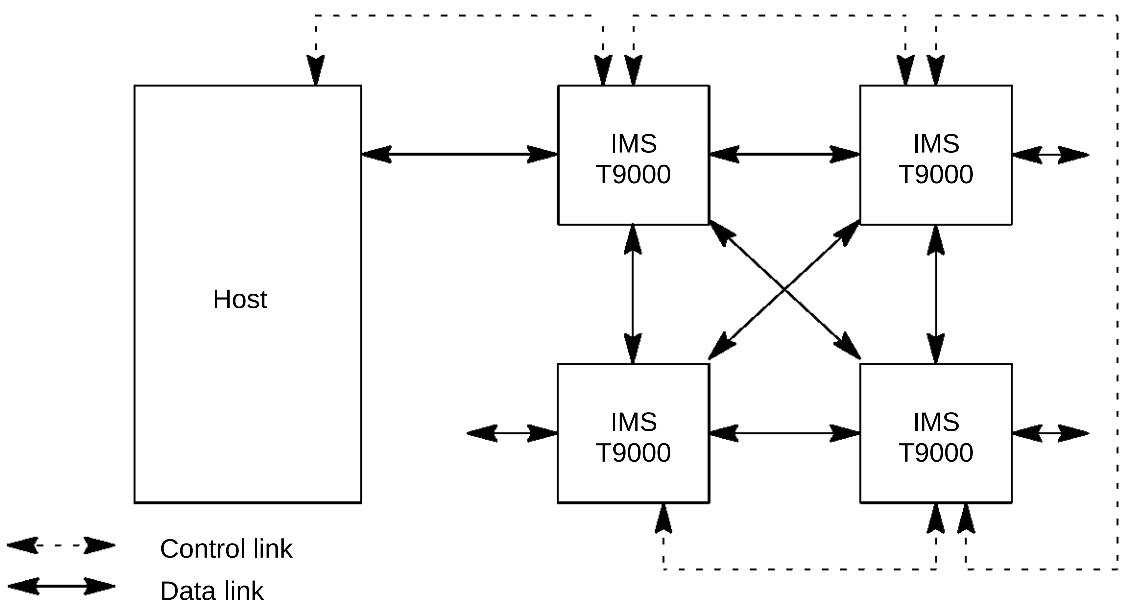


Figure 2 IMS T9000 network example

This hardware multiplexing support is complemented by a software through-routing mechanism provided by the T9000 occam 2 Toolset. The combination of hardware multiplexing and software through-routing allow any user-defined channel connections to be specified and implemented with optimal flexibility.

The STC104 packet routing switch gives higher communications performance in larger systems. It is a fast packet router which can route each incoming packet out along any one of its links, depending on the packet header added to the data by the sending transputer. It enables direct connections between the transputers in a large network and can remove the need for software through-routing. Large networks may also be implemented without packet routing switches, in which case each IMS T9000 may have a direct connection to up to four other devices.

An IMS T9000 transputer system has a network of links for data communications plus a separate network of control links for system initialization and debugging control. An IMS T9000 network can be viewed as a number of processors each of which is connected to both the data network and the control network. During system development, both networks should also be connected to the host or other controlling processor, as shown in Figure 1.

2.1 The IMS T9000 family of devices

The T9000 occam 2 Toolset supports the full family of the IMS T9000 transputer and the associated STC104 packet routing switch. The STC104 packet routing switch is a 32-way packet router.

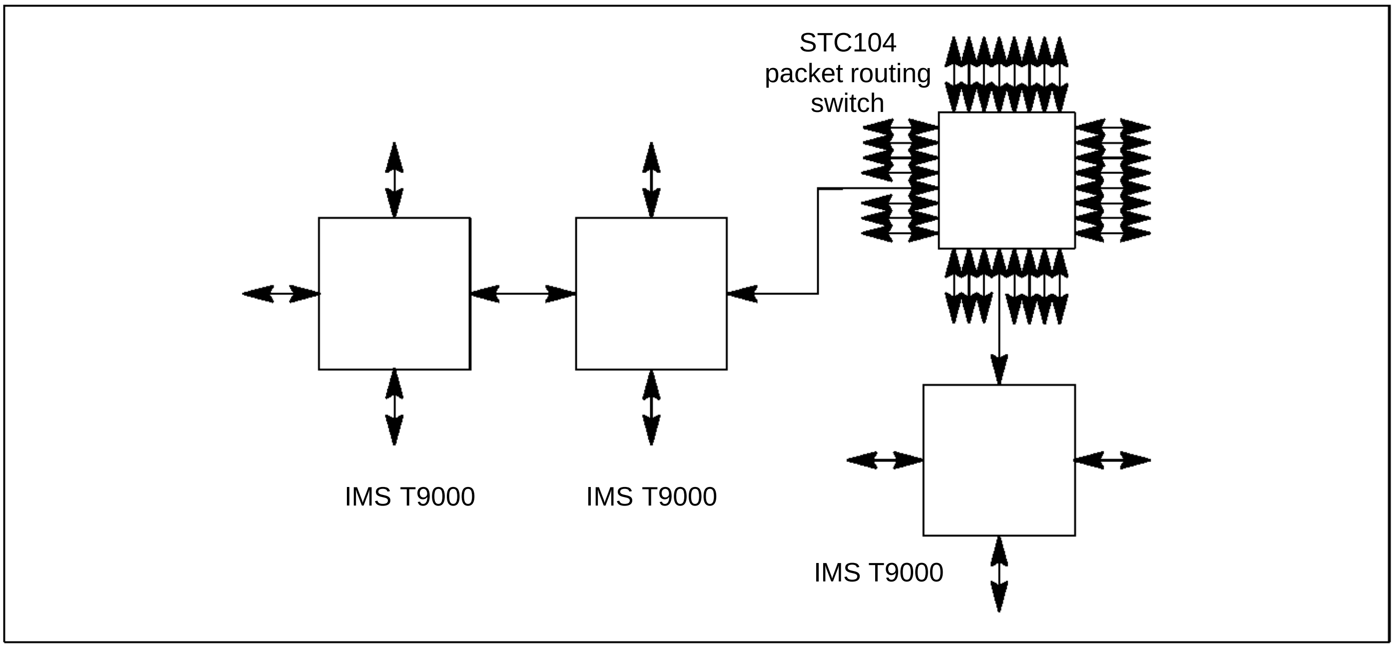


Figure 3 IMS T9000 family of devices

3 Parallel programming

The T9000 occam 2 Toolset supports parallelism on individual transputers and parallelism across networks of IMS T9000 transputers.

The transputer programming model consists of parallel processes communicating through channels. Channels connect pairs of processes and allow data to be passed between them. Each process can itself be built from a number of parallel sub-processes, so that an entire software system can be described as a process hierarchy. Processes may be created at high and low priority levels. Interrupt routines are typically implemented as high priority processes. This model is consistent with many modern software design methodologies.

occam 2 is an efficient and effective language for parallel programming. It is a high level, structured language, providing simple constructs for creating parallel processes and for message-passing using channels. Each channel has a `PROTOCOL` which determines the types of messages that may flow along it. The occam compiler provides comprehensive compile and run-time checking which considerably reduces the number of coding and usage errors.

Figure 4 shows a collection of four processes communicating through channels. The `mux` process communicates with a host computer and hands out work to be done to one of three `worker` processes. Results from the workers are then returned to the host by the `mux` process. The following example shows how this collection of processes can be described in occam to run on a single IMS T9000 transputer. Section 6.4 shows how the parallel processes in the program can be mapped onto a network of processors using the configuration language.

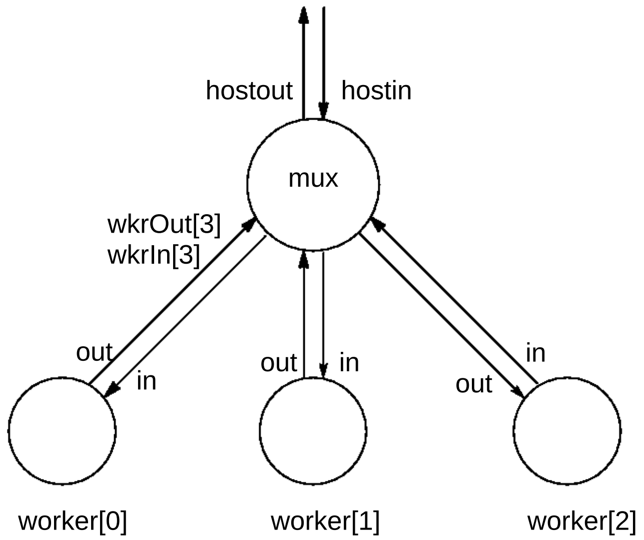


Figure 4 Software network

The code in Figure 5 illustrates how to program the collection of parallel processes shown diagrammatically in Figure 4. The files `mux.tco` and `worker.tco` contain the executable code of the processes `mux` and `worker` respectively. These processes communicate using channels.

```
#INCLUDE ”hostio.inc” –– contains SP protocol definition
#USE ”mux.tco”
#USE ”worker.tco”

PROC example (CHAN OF SP hostinput, hostoutput, []INT memory)
  [3]CHAN OF SP wkrIn, wkrOut:
  PAR
    mux(hostinput, hostoutput, wkrOut, wkrIn)
    PAR i = 0 FOR 3
      worker(wkrIn[i], wkrOut[i])
:
```

Figure 5 Parallel processes

The occam 2 compiler in the Toolset can be used to compile the program shown in Figure 5. The compiled code can then be linked, configured and collected to produce a code file to run on a single transputer. Alternatively it may be preferred to distribute the processes over a network of more than one processor, in which case the code for the `mux` and `worker` processes must be compiled and linked separately. The linker produces processes in the form of fully linked units. A network of fully linked processes can be distributed over a network of processors using the configuration tools. The code shown in Figure 5 would be replaced by a configuration description, as described in Section 6.4.

occam constructs are provided to read a message from one of a list of channels, to time-out on channel input and to access the high and low resolution timers built into the IMS T9000 transputer. The transputer's hardware scheduler provides extremely efficient scheduling of these processes and efficiently implements many features which would normally require a real-time executive.

The compiler operates from a host command line interface. The compile time diagnostics provided by the compiler are excellent. These include type checking in expressions and type checking of function arguments.

The tools integrate into the host operating system build utilities, allowing, for example, the use of standard editor, make and source code and configuration control utilities.

4 Compiler and compilation tools

4.1 Optimization

The compiler implements a wide range of code optimization techniques.

Constant folding. The compiler evaluates all constant expressions at compile time.

Advanced workspace allocation. Frequently used variables are placed at small offsets in workspace, thus reducing the size of the instructions needed to access them and ensuring that the workspace cache of the IMS T9000 is employed for frequently used variables. This therefore increases the speed of execution.

Constant caching. Some constants have their load time reduced by placing them in a constant table.

CASE statements. The compiler can generate a number of different code sequences tailored to cover the dense ranges within the total range.

Special idioms that are better on transputers are chosen for some code sequences.

In-line code Procedures and functions can optionally be implemented as in-line code.

The compiler and linker provide features which allow programmers to make good use of the IMS T9000 transputer's internal memory when used as on-chip RAM.

4.2 Use of IMS T9000 features

Programs are compiled to run as L-processes (with a local trap handler) on the IMS T9000. A trap handler is set up to catch and handle program runtime errors. The behavior of the program after an error (HALT or STOP) is determined by the trap handler installed rather than at compilation time.

The code generated by the compiler and in the supporting libraries, makes full use of the new features introduced to the transputer instruction set in the IMS T9000. In particular:

- The part-word support instructions are used to improve the handling of 8-bit and 16-bit integers.
- The new floating point instructions, *fpsqrt* and *fprem* are used.

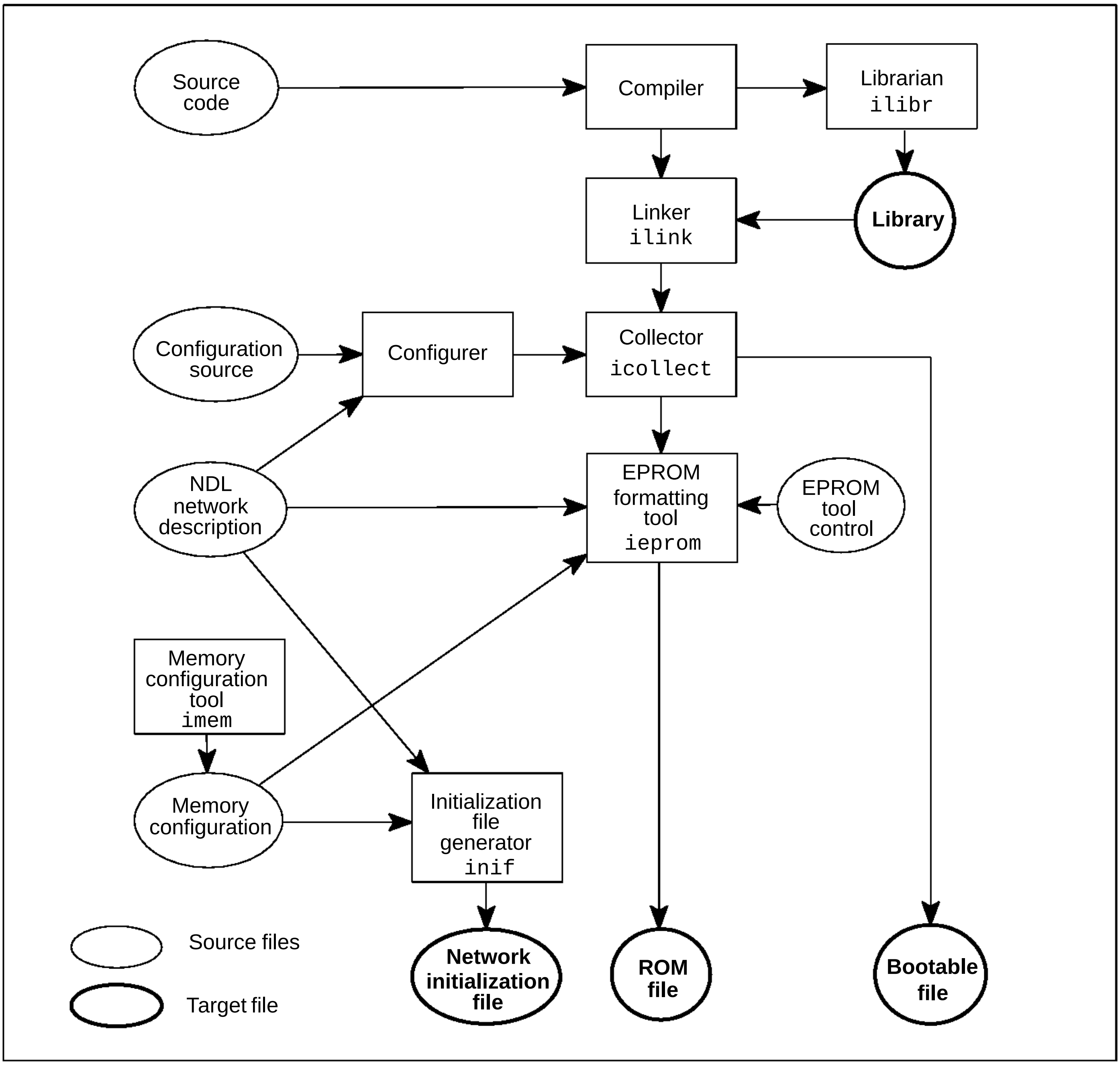


Figure 6 Program build model for the T9000 occam 2 Toolset

- PORTs are implemented using the device access instructions.
- The semaphore support in the runtime library is implemented using the new semaphore instructions.
- Counted protocols are implemented using the variable length I/O instructions *vin* and *vout*.
- Care has been taken in the code generation and optimization to ensure that effective use is made of the IMS T9000's workspace cache and CPU pipeline.

Access to the full IMS T9000 instruction set is available through assembly inserts.

4.3 TCOFF

The binary code produced by T9000 occam 2 Toolset tools is in Transputer Common Object File Format (TCOFF). This allows integration with other TCOFF-based utilities, including the T9000 ANSI C Toolset and the INQUEST development environment.

4.4 Separate compilation

Collections of subprograms can be compiled separately using the occam 2 compiler and optionally combined into a library. The linker is used to combine separately compiled functions and procedures as a linked unit. A single copy of a linked unit cannot be distributed over more than one processor. The linker supports selective loading of library units.

4.5 Mixed language programming

The T9000 ANSI C and occam 2 compilers are fully compatible and allow simple mixing of languages. occam and C processes may be freely mixed when configuring a program for a single transputer or a network of transputers. Such processes will run in parallel and communicate using channels.

The T9000 occam 2 Toolset supports calling C functions directly from occam just like other occam procedures. Pragmas are provided to tell the compiler to generate the hidden static link parameter (which is required by C but not by occam) and to change the external name of a function, since C functions may have names which are not legal occam procedure names.

Similarly, the T9000 ANSI C Toolset allows occam 2 procedures and single-valued occam 2 functions to be called from C just like other C functions.

4.6 Assembler code

The T9000 occam 2 Toolset provides a very powerful assembler insert facility. Assembler code can be written at any point in the source code to achieve direct access to transputer instructions for reasons of speed or code size. Full access is available to the IMS T9000 transputer instruction set and occam program variables.

The assembler insert facility supports:

- Access to the full instruction set of the IMS T9000 transputer
- Symbolic access to variables
- Pseudo-operations to load multi-word values into registers
- Loading results of expressions to registers using the pseudo-operations.
- Labels and jumps
- Directives for instruction sizing, stack access, return address access etc.

5 Libraries

The T9000 occam 2 Toolset provides a wide range of occam libraries, including mathematical functions, string operations and input/output functions. The libraries support similar functions across the full range of transputer types, making it easy to port software between transputer types. Sources of most of the libraries are provided, for adaptation if required.

The libraries provided are listed below.

occam compiler library

This is the basic occam run-time library. It includes: multiple length arithmetic functions; floating point functions; IEEE arithmetic functions; 2D block moves; bit manipulation; cyclic redundancy checks;

code execution; arithmetic instructions. The compiler will automatically reference these functions if they are required.

snglmath.lib, dblmath.lib

Single and double length mathematics functions (including trigonometric functions). These libraries use floating point arithmetic and will produce identical results on all processors. The occam source code is also provided.

string.lib

String manipulation procedures. The occam source code is also provided.

hostio.lib

Procedures to support access to the host terminal and file system through the file server. The occam source code is also provided.

streamio.lib

Procedures to read and write using input and output streams. The occam source code is also provided.

msdos.lib

Procedures to access certain DOS specific functions through the file server. The functions are specific to the IBM PC. The occam source code is also provided.

crc.lib

Procedures to calculate the cyclic redundancy check (CRC) values of strings.

convert.lib

Procedures to convert between strings and numeric values.

xlink.lib

Procedures implementing link communications which can recover after a communication failure.

6 Configuration, initialization and loading

Configuration is the process of defining how an application program is to be run on the available hardware. Given a description of the hardware network, the software network of an application and the mapping between them, the T9000 occam 2 Toolset produces a *bootable file* which can be sent to the network for execution. The hardware network description is also used to prepare a *network initialization file* which is used to initialize the network and bootstrap the processors into a state where they are ready to receive the bootable file.

A configuration description file is used as input to a tool known as the *configurer*. The description is written in an occam-like language which is upwards compatible with the configuration language in the Dx305 Professional occam 2 Toolset. The configuration description describes the application as a network of processes and channels. It indicates which linked files should be used as the code for each process, and it also indicates on which processor each process is to be run.

The hardware description is separate from the rest of the configuration description. It is written in the Network Description Language (NDL). The configuration description refers to the NDL description of the hardware by means of a #NETWORK directive. The hardware description describes the processors and routing devices in the network and their connections.

The main features of the configuration tools are as follows:

- To initialize, boot and load an IMS T9000 network, it is not necessary to write any low-level code; instead the attributes of devices are specified by simple textual statements in the network description, and the low level code is generated automatically by the tools.
- The network description for a particular network can be written once, when the hardware is designed, and does not normally need to be changed between different applications.
- The configuration tools will check the correctness of the network description; this includes checking the labelling of the routing chips for deadlock freedom.
- Channels specified by the user in the configuration description are automatically mapped by the tools. In a system with routing devices, the configuration tools calculate the routing headers required for the user's channels. If there are no routing devices in the system, the configurer adds any software through-routing mechanisms required. Thus the software network is not constrained by the topology of the hardware network.
- For a particular application, the tools can produce either a bootable file which can be used to load the network from a host, or a boot-from-ROM file which can be used to program a ROM on one of the processors in the network.
- Initialization of individual IMS T9000 devices in the network (for example, setting up the external memory interface) can be done over the control link, or from an initialization ROM local to each IMS T9000. Support for generation of local initialization ROMs is included.

6.1 Network description

The Network Description Language (NDL) is used to describe the available hardware – the types of processors, their attributes and how they are connected. Processor attributes include, for example, a description of its memory map and its link speeds. The NDL also describes any packet routing switch devices in the network and their attributes. Attributes of a routing device include the *labelling* of the routing device, which indicates how packets from processors should be routed through it.

The network description will not normally change unless the hardware is changed. This NDL description of the system is used by the tools for a variety of purposes, from initializing the hardware to mapping application code onto processors. These tools can either read and check the NDL source directly or read a binary version produced by the NDL compiler, `indl`.

Figure 7 shows the network description for the example network shown in Figure 10.

6.2 Memory configuration

The IMS T9000 programmable memory interface supports a wide variety of memory configurations. It can be set up to provide the appropriate signals and timing needed by the memory being used. The parameters to initialize the memory interface may be sent to the transputer using the control link or may be included in the bootstrap code in ROM.

```
#INCLUDE "stdndl.inc"

-- Name of file with memory timing information, etc.
VAL memconfig IS "T9000.mem" :


VAL n       IS 4 :                        -- Number of transputers
VAL SysMem  IS 2*K :                      -- Size of system RAM
VAL UsrBase IS #80000000 :                -- Start address of user RAM
VAL UsrMem  IS (2*M) - SysMem :           -- Size of user RAM
VAL SysBase IS UsrBase+UsrMem :           -- Start of system RAM
                                          -- (= top of user RAM)
VAL memoryflags IS [[SysBase, SysMem, RAM + SYSTEM],
                    [UsrBase, UsrMem, RAM + USER]] :


[n]NODE p :                               -- n transputers
CONTROLPORT host :
ARC         hostlink :


NETWORK fourT9
  DO
    -- set link speeds
    SET DEFAULT (link.speed.multiply  := 10)
    SET DEFAULT (link.speed.divide    := [1])
    SET DEFAULT (control.speed.divide := [8])

    -- set processor types
    DO i=0 FOR n
      SET p[i] (type := "T9000")
    SET p[0] (root := TRUE)
    DO i=1 FOR n-1
      SET p[i] (root := FALSE)

    -- set memory information
    DO i=0 FOR n
      SET p[i] (memconfig, memory := memconfig, memoryflags)

    -- connect control network
    CONNECT host[control] TO p[0][control.up]
    DO i=0 FOR n-1
      CONNECT p[i][control.down] TO p[i+1][control.up]

    -- connect data links
    CONNECT host[data]  TO p[0][link][0] WITH hostlink
    DO i=0 FOR n-1
      CONNECT p[i][link][1] TO p[i+1][link][2]
    CONNECT p[n-1][link][1] TO p[0][link][2]
    DO i=0 FOR n-2
      CONNECT p[i][link][3] TO p[i+2][link][0]
:
```

Figure 7 NDL network description example

The memory configuration tool, `imem`, is used to assess and define memory interface parameters for each IMS T9000. It can be run interactively or in batch mode. The output may be incorporated into the network initialization file or into a ROM. The memory configuration description will not normally change unless the hardware is changed. `imem` can also be used to generate memory interface timing documentation.

6.3 Initializing and loading

A hosted IMS T9000 network is initialized by first sending control commands to the transputers and routers connected to the control network and then loading code using the data network.

The correct sequence of control network commands is held in a network initialization file, which contains all the information needed to initialize a system through the control links prior to loading the application code. It can be automatically generated by the initialization file generator tool, `inif`, from the network description and memory configurations. To initialize the network, the network initialization file is used by the initialization software to generate the correct sequence of commands to send to the control link network.

The bootstrap code for each processor, loading code and user application code are all incorporated in the bootable file with the necessary routing information. The bootable file is generated automatically by the collector, `icollect`, from the configuration information and linked application code files. To load the code onto the network, the bootable file is sent down the data link to the data link network.

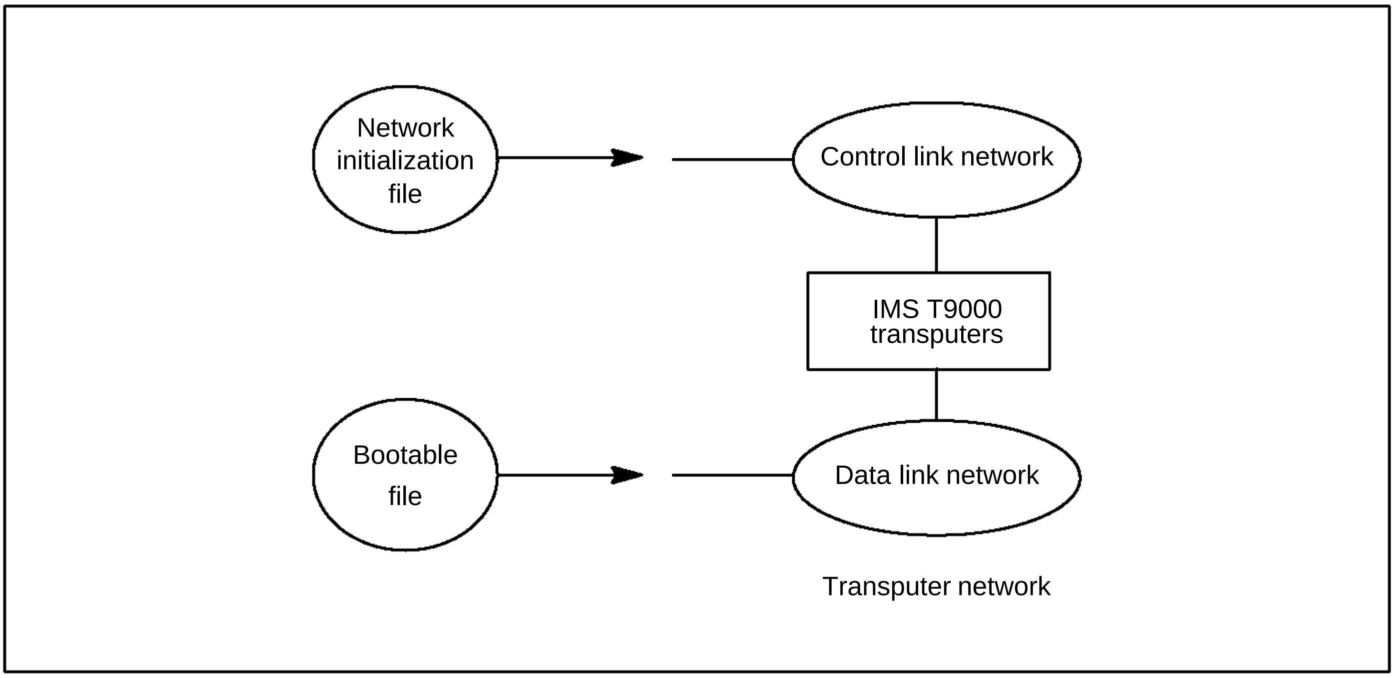


Figure 8 Initializing a hosted IMS T9000 network

6.4 Configuration language

An occam-like configuration language is used to describe the network of processes and channels and the mapping of the processes onto the transputer network. The transputer network is described separately in the network description file. Multiple processes may be mapped onto the same transputer. The network description file is referred to by means of a `#NETWORK` configurer directive. This allows the user to map processes in the configuration description onto the processors named in the network description file. The routing of channels may be generated automatically or may be included in the configuration description.

The following example illustrates just how easy it is to configure a program for transputers. Instead of using the top level single transputer code shown in Figure 5, it may be preferred to distribute the program over a network of processors, as shown in this example.

```
#NETWORK ”fourT9.ndl”

#INCLUDE ”hostio.inc”
#USE ”mux.lku”
#USE ”worker.lku”

CONFIG
  CHAN OF SP hostin, hostout:
  PLACE hostin, hostout ON hostlink:
  [3]CHAN OF SP wkrIn, wkrOut:
  PAR
    PROCESSOR p[0]
      mux(hostin, hostout, wkrOut, wkrIn)
    PAR i = 0 FOR 3
      PROCESSOR p[i+1]
        worker(wkrIn[i], wkrOut[i])
:
```

Figure 9   Configuration

Figure 9 shows the configuration text for describing the software network of processes and channels and mapping the software onto the hardware shown in Figure 10 and described in NDL in Figure 7. The `mux` and `worker` processes in the software network have been compiled and linked into the files `mux.lku` and `wkr.lku` respectively. The NDL shown in Figure 7 is in the file `fourt9.ndl`. The software description in the configuration text replaces the top level of code shown in Figure 5. In this example the `mux` process runs on the root transputer, called `p[0]`, while the individual `worker` processes run one on each of the transputers labelled `p[1]` to `p[3]` in Figure 10.

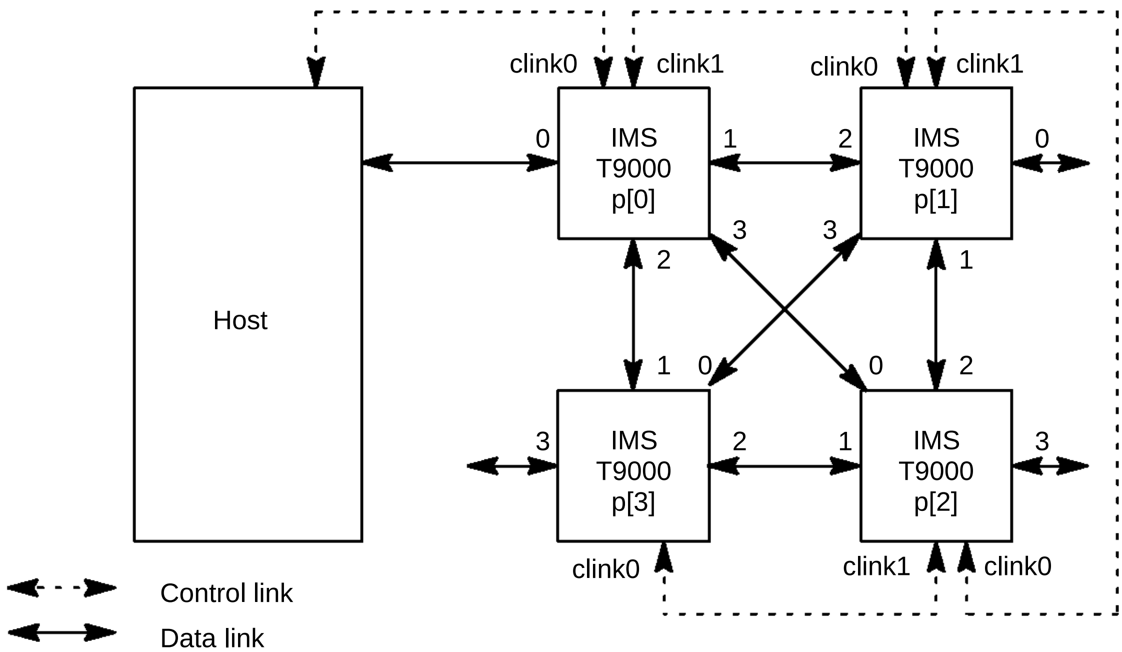


Figure 10   IMS T9000 network example showing link numbers

6.5   ROM support

The software can usually be developed and tested in a hosted development system without ROM by using host files loaded via transputer links. As a final stage, the completed application software and initialization data can be loaded into EPROMs using the EPROM program formatting tool, `ieprom`.

The EPROM formatting tool can produce a file suitable for programming a ROM; this may be either a system ROM or a local ROM. A system ROM holds network initialization data and application software for initializing and loading onto a network of transputers. Stand-alone systems will need to boot from a system ROM. A local ROM holds local initialization data for only one transputer in a network. One or more transputers in a network may be initialized from local ROMs, whether the application is loaded from a host file or from a system ROM.

The EPROM formatting tool may read the network description, memory configurations and the collected application software. From this data, the EPROM formatting tool can build a ROM file incorporating the network initialization data and application software. A control file allows the user to control how the data is arranged in the ROM. The ROM file may be produced in ASCII hexadecimal, Intel hexadecimal, extended Intel hexadecimal, Motorola S-record or binary format.

7 Product components

7.1 Tools

occam 2 compiler, linker and librarian – `oc`, `ilink`, `ilibr`

Makefile generator, binary lister program and memory map lister – `imakef`, `ilist`, `imap`

Configuration tools – `indl`, `imem`, `inif`, `occonf`, `icollect`

EPROM programming tool – `ieprom`

7.2 Libraries

Mathematics functions (includes sin, cos, etc.) – `snglmath.lib`, `dblmath.lib`

String manipulation procedures – `string.lib`

File and terminal i/o procedures – `hostio.lib`, `streamio.lib`

Access to certain MS-DOS calls – `msdos.lib`

CRC calculation procedures – `crc.lib`

String / number conversion procedures – `convert.lib`

Extraordinary link handling procedures – `xlink.lib`

Mixed language support library – `callc.lib`

Configuration support libraries

7.3 Sources

Programming examples

Configuration support library source files

Network description examples

Network control software source files

occam libraries source files

7.4 Documentation

T9000 occam 2 Toolset User Guide

T9000 Toolset Reference Manual

T9000 occam 2 Language and Libraries Reference Manual

T9000 Toolset Hardware Configuration Manual

T9000 Delivery Manual

T9000 occam 2 Toolset Handbook

occam 2 Language Reference Manual

Tutorial Introduction to occam

8 Product variants

8.1 Sun-4 product

- IMS D4395 T9000 occam 2 Toolset

Operating requirements

For Sun-4 hosted cross-development the following will be required:

- A Sun-4 workstation or server with 1/4 inch tape drive capable of reading QIC-24 format;
- SunOS 4.1.3 or Solaris 2.4 or compatible
- 9 Mbytes of free disk space.

For loading target systems, a suitable transputer network interface will be required, such as an IMS B103 Ethernet to DS-Link Interface board and IMS S4397 Sun 4 T9000 Network Interface Software.

Distribution media

Sun-4 software is distributed on DC600A data cartridges 60 Mbytes, QIC-24, tar format.

Licensing

The IMS D4395 T9000 occam 2 Toolset is supplied with a single-user license. The tools can be run on any Sun-4 machine that is part of a network connected to a single machine where the licence manager is installed. Further information about the licence manager is included in the product Delivery Manual. Multiple copies can be purchased for larger project teams using volume discount curves.

No licence fee is charged for including libraries in customer products when linked with customer applications using the linker `ilink`. Example programs and other sources provided may be included in software products, but SGS-THOMSON Microelectronics Limited retain original copyright. Full licensing details are available from SGS-THOMSON Sales Offices, Regional Technology Centers and authorized distributors.

8.2 PC product

- IMS D7395 T9000 occam 2 Toolset

Operating requirements

For PC hosted cross-development the following will be required:

- IBM 386 PC (or compatible) with a minimum of 8 Mbytes memory
- DOS 5.0 or later
- 9 Mbytes of free disk space

For loading target systems, a suitable transputer network interface will be required, such as an IMS B108 add-in board or IMS B103 Ethernet to DS-Link Interface board and IMS S7397A PC T9000 Host Interface Software.

Distribution media

Software is distributed on 1.44 Mbytes 3.5 inch IBM format diskettes.

Licensing

The IMS D7395 T9000 occam 2 Toolset is a single-user product. Multiple copies can be purchased for larger project teams using volume discount curves.

No licence fee is charged for including libraries in customer products when linked with customer applications using the linker, `ilink`. Example programs and other sources provided may be included in software products, but SGS-THOMSON Microelectronics Limited retain original copyright. Full licensing details are available from SGS-THOMSON Sales Offices, Regional Technology Centers and authorized distributors.

9 Problem reporting and field support

A registration form is provided with each product. Return of the registration form will ensure you are eligible for future product updates. Software problem report forms are included with the software. SGS-THOMSON products are supported worldwide through SGS-THOMSON Sales Offices, Regional Technology Centers and authorized distributors.

10 Ordering information

Description	Order number
T9000 occam 2 Sun 4 Toolset.	IMS D4395
T9000 occam 2 PC Toolset.	IMS D7395

Table 1 Ordering information

SGS-THOMSON Microelectronics GROUP OF COMPANIES

Australia - Brazil - Canada - China - France - Germany - Hong Kong - Italy - Japan - Korea - Malaysia - Malta - Morocco
The Netherlands - Singapore - Spain - Sweden - Switzerland - Taiwan - Thailand - United Kingdom - U.S.A.