

**Network Configuration Manager
Software Documentation
Copyright (C) 1990, Parsytec GmbH**

Authors:

H. Blanke-Bohne

M. Sigg

S. Staas

**Network Configuration Manager
for MultiCluster 2 and SuperCluster
User Documentation**

Version 3.0, December 1990

Copyright (C) 1990 by Parsytec GmbH

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Parsytec GmbH, W-5100 Aachen, Juelicher Strasse 338, Federal Republic of Germany.

Macintosh and MAC II are the trademarks of Apple Computer Inc.

UNIX is a trademark of AT&T Bell Laboratories.

VAX is a trademark of Digital Equipment Corp.

inmos and occam are trade marks of the INMOS Group of Companies.

IBM, PC-AT are trademarks of International Business Machines Corp.

Helios is a trademark of Perihelion Software Ltd.

SUN is a trademark of Sun Microsystems, Inc.

Printed in Germany

Contents

Contents	4
Preface	7
Preface to Version 3.0	9
1 The Hardware	11
1.1 Electronically Reconfigurable Systems	12
1.2 Available Transputer Systems	15
1.2.1 Fixed Topology	15
1.2.2 Logical Reconfigurability	16
1.2.3 Physical Reconfigurability	16
1.3 Network Reconfiguration	18
1.3.1 MultiCluster 2/16-1 NCU and SuperCluster 16	18
1.3.2 MultiCluster 2/32-2 NCU and MultiCluster 2/64-4 NCU	20
1.3.3 SuperCluster	26
1.4 System Upgrade and Expandability	34
1.4.1 MultiCluster 2	34
1.4.2 SuperCluster	35
2 The System Software	36
2.1 Operation of the System Software	37
2.1.1 User Requests	37
2.1.1.1 Ringing the NCM	38
2.1.1.2 Network Configuration Requests	39
2.1.2 Dispatcher Requests	42
2.2 Running and Using the System Software	43
2.2.1 MultiTool »NCM.control«	45
2.2.1.1 Booting	45
2.2.1.2 Error Situations and Warnings	49
2.2.1.3 The NCM Monitor	51

2.2.2 Helios »boot ncm«	60
2.2.2.1 Booting	60
2.2.2.2 Error Situations and Warnings	63
2.2.2.3 NCM Control Functions	66
2.2.2.4 Exit Codes	68
2.2.3 ConfigIn Link Protocol	69
2.2.3.1 Booting	69
2.2.3.2 NCM Control Functions	71
3 The User Interface Software	74
3.1 User Interface Software for MultiTool	76
3.1.1 Machine Dependent Network Configuration	76
3.1.1.1 MultiCluster 2/16-1 NCU and SuperCluster	77
3.1.1.2 MultiCluster 2/32-2 NCU and /64-4 NCU	78
3.1.2 The »EXE Config and Boot«	79
3.1.3 The Extended Occam 2 Compiler Utilities	98
3.1.4 Example PROGRAMs for MultiTool	105
3.2 User Interface Software for Helios	106
3.2.1 Hardware Connection to the MultiCluster 2 / SuperCluster	107
3.2.2 Manual Usage of Helios »hconfig«	107
3.2.3 Automatic Usage of »hconfig« on PC Hosted Systems	112
3.2.4 Automatic Usage of »hconfig« on SUN Hosted Systems	113
3.2.5 Setting Up the Resource Map	114
3.2.6 Syntax of the Resource Map (MC 2/32 and /64)	115
3.2.7 Some Useful Hints	117
3.2.8 Error Messages	118
3.2.9 Exit Codes	121
3.2.10 Example Resource Maps for Helios	122
3.3 Special Requirements for Processor Numbering	126
3.4 Connecting of Special Processor Boards	128
3.5 User Entries	131
3.5.1 Processor Connected to the First User Entry	131
3.5.2 The Second User Entry (for SuperCluster only)	132
3.5.3 Available User Entries	133
3.6 NCM User Interface Protocol	134
3.6.1 Ringing the NCM	134
3.6.2 Sending a Request	135
3.6.3 Syntax of Network Descriptor	141

4 Installation	142
4.1 Installation on a PC	142
4.1.1 System Software for MultiTool	143
4.1.2 System Software for Helios	143
4.1.3 User Interface Software for MultiTool	144
4.1.4 Extended Occam 2 Compiler (MC 2/32 and /64)	146
4.1.5 User Interface Software for Helios	146
4.2 Installation on a MultiCluster 1 Unit	147
4.3 Installation on a SUN Workstation	148
4.3.1 Main Installation	148
4.3.2 Extended Occam 2 Compiler (MC 2/32 and /64)	149
4.3.3 User Interface Software for MultiTool	149
4.3.4 User Interface Software for Helios	150
A Quick Reference	151
A.1 Running and Using the System Software	151
A.2 Running the User Interface Software	153
A.2.1 Configure a Network and Load a MultiTool User Program	153
A.2.2 Configure a Network for Booting Helios	155
B Glossary	156
C Error Numbers	170
D Physical Device Numbers	188
D.1 Physical Numbering of NCUs and Application Processors	188
D.2 Physical Numbering of Special Processors	196

Preface

Till today PARSYTEC has developed a number of *electronically reconfigurable machines* for the powerful concept of communication oriented parallelism.

These machines are scalable for any number of transputers and suitable for multiuser environments.

One of the highlights of PARSYTECs system design is the flexibility and expandability based on a homogeneous and simple system structure.

Control and communication between its various components is uniformly achieved through the links by message passing, extending the transputer concept to higher levels. No central control bus is involved, which would interfere with the homogeneous structure and which would cause conceptional limitations concerning expansion capabilities.

Electronically reconfigurable machines

The heart of all electronically reconfigurable machines and of all multiuser capabilities is a *configuration control system* which is built up with a number of *Network Configuration Units (NCUs)*. The NCU's processors form a transputer network where the system software, the Network Configuration Manager (NCM), is running. Furthermore each NCU contains two crossbar switches of a capacity of 96 x 96 UniLinks to connect the links of the system's transputers.

The number of NCUs required depends upon the number of processors in the system and the system's reconfigurability.

Thus, the main criteria to classify a transputer system are *reconfigurability* and *multiuser capabilities*.

The *MultiCluster 2* can be configured to cover the broad range from fixed topology single user systems to reconfigurable multiuser systems.

This is achieved by installing Fixed Topology Modules (FTMs) or NCUs to support a maximum of 64 application processors.

The *SuperCluster* is the highest level of reconfigurable multiuser systems. A system with 320 processors was installed in December 1990 at the University of Paderborn. Larger machines may follow.

Network Configuration Manager (NCM)

Another highlight is the unique system software, according to its task called the *Network Configuration Manager* or short NCM. The NCM offers a full support of the machine's flexibility and expandability and a unique surface for the users of the various reconfigurable systems.

This operating system is running on all electronically reconfigurable machines. It will help the dispatcher to maintain the system with status reports and test features.

The other functions of the NCM are reached via the

User interface software

The different host systems with their software environments will be supported by the *user interface software* which may communicate with the NCM to request transputer networks and status reports.

About this manual

As prerequisites, we assume the reader is familiar with MultiTool or Helios respectively at least on a basic level.

Chapter 1 provides a general overview of the different MultiCluster 2 and SuperCluster models. The main hardware structures are described and their relation to the operating software is presented.

Chapter 2 explains the system software and chapter 3 the user interface software in a comprehensive way.

Chapter 4 describes the installation of the software packages on different host systems.

Appendix A gives a summary of the different actions to proceed in order to use the system and user software.

Appendix B contains a glossary which explains the different expressions used in this document in detail.

Appendix C is a collection of error numbers.

Appendix D shows some details on physical processor numbering.

Preface to Version 3.0

Three years ago, the first reconfigurable machine, a SuperCluster 64, has been developed by PARSYTEC in cooperation with Dr. Mühlenbein, GMD (Society for Mathematics and Computer Science in the Federal Republic of Germany).

The first version of the NCM was programmed by Reinhard Rinn, now Vice President of Parsytec Inc. in West Chicago (USA).

Since then, SuperClusters with a less number of processors (32, 16 and 48) were developed and a corresponding NCM (version 2.2) supported all these machines and was able to integrate 2 special processors.

At the end of 1989, PARSYTEC placed his MultiCluster 2 series on the market. These machines were designed in cooperation with Prof. Monien of the University of Paderborn.

For each machine a NCM was developed (MC 2/16 : version 1.0, MC 2/32 : version 1.1, MC 2/64 : version 1.2).

In April 1990, the first SuperCluster with more than 64 processors, the SuperCluster 128, was installed at the University of Heidelberg in the Federal Republic of Germany. 128 processors could be connected in any topology by the NCM version 2.3. The partitioning algorithm was developed in cooperation with Reinhard Lüling who is working with Prof. Monien at the University of Paderborn, where a SuperCluster with 320 transputers is available today.

Since the last NCM version for SuperCluster 64, much development has been done. The SuperCluster, itself developed for MultiTool, has become a Helios machine, too.

With the NCM version 3.0, all the different versions and their development come together. A unique NCM for all MultiCluster 2 and SuperClusters may configure networks in the same way for all machines.

Error Codes from the NCM may inform the users why their configuration failed, the synchronisation protocol for ringing was improved, communication between the NCU processors was secured (you may interrupt the NCU's main processor pipe without disturbing the NCM), and a startup request will detect a running NCM at the ConfigIn link. (Startup requests are used in »NCM.control«, »boot_ncm«, »Config and Boot« and »hconfig«.)

The administration was extended to the NCU channels. All free channels into a Computing Cluster may be used independently of the number of processors allocated in the cluster. This solves the problem of a 7x7 torus on the SuperCluster 64.

Some more special processors are supported by the NCM (e.g. 8 special processors for SuperCluster 64 instead of 2 in version 2.2).

Furthermore, the NCM becomes loadable from Helios by the new tool »boot_ncm«.

The »hconfig« supports special processorboards, delivers exit codes for batches and produces textual error messages from the NCM error code. A status request (free processors and special processors, processor usage, mapping of user processors to physical system processors and NCU connections for the user) is realised in »hconfig«.

For link renumbering (MultiCluster 2) the syntax check for the .rm resource map file has been improved.

For »Config and Boot« the graph detected by the NCM (also in case of error) can be displayed by a new function <d>. NCM error codes are translated into textual output.

Synchronisation failure while ringing leads to timeout. Ringing at ConfigIn will be detected.

A lot of improvements and extensions have been inserted in the NCM system interface used by the »NCM.control«:

Connecting channels will not disturb user networks, »disconnect channels« and »reset NCUs« will produce messages if a user network was touched.

The physical status request for NCU connections is divided into 'link level' and 'reset level' corresponding to the two boards of each NCU. 'User network return' is possible via the dispatcher now.

Usage of resources (processors, special processors and NCU channels) can be shown via the interactive system interface.

Last not least, this documentation will help you to use the Network Configuration Manager to configure any network you need for your application.

Aachen, December 1990

Helmut Blanke-Bohne

1 The Hardware

First of all this document is a software manual. Nevertheless some knowledge of the systems running this software is necessary and useful. Thus, this chapter will introduce the different hardware models from the software's point of view. Additionally some information about expandability of the systems is given.

The most important fact to mention is that only the electronically reconfigurable machines (MultiCluster 2 with at least one NCU or SuperCluster) need to run the software described in chapters 2 and 3.

You will find a more sophisticated presentation of the machines' hardware in the particular Technical Documentations.

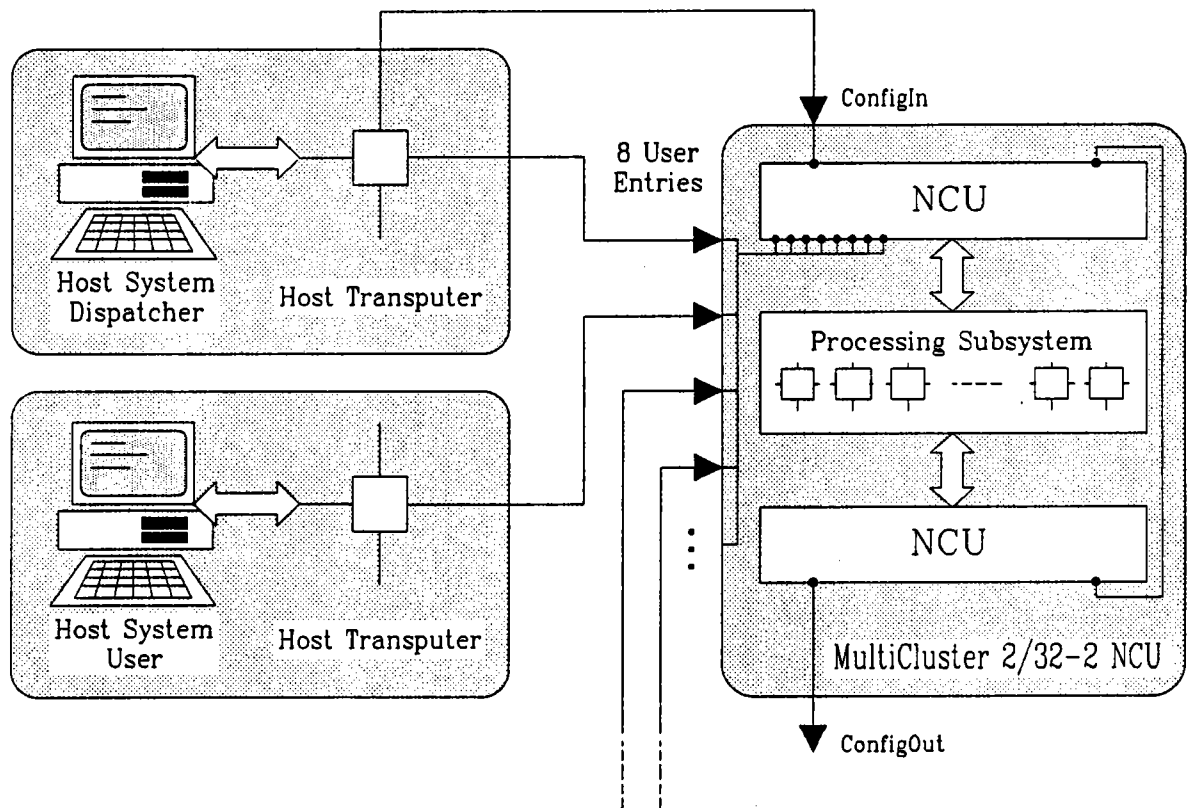


Fig 1.1 Connection host to MultiCluster 2/32-2 NCU

1.1 ELECTRONICALLY RECONFIGURABLE SYSTEMS

An electronically reconfigurable system mainly consists of three components: a *processing subsystem* with the application transputers, an *I/O subsystem* of special processors (mass storage-, graphics-, video-controllers, Ethernet interface, etc.), interface boards for industry and research applications and a *configuration subsystem* for the general resource management.

The configuration subsystem manages the processing subsystem and the I/O subsystem which represent the resources (transputer and periphery) of the machine. Via the configuration subsystem the transputer links can be switched to configure a network needed for running an operating system or loading a user application.

Special slots are reserved to integrate *Network Configuration Units (NCUs)*. A certain number of these boards represent the hardware of the entire configuration subsystem, the heart of any electronically reconfigurable machine developed by PARSYTEC. Due to the importance of this system part we will have a closer look at it in the following section.

The Network Configuration Unit (NCU)

Each NCU contains two transputers, a *main processor* (IMS T414 or IMS T800) and a *sub processor* (IMS T212). These transputers are for management purposes only and are not available for application tasks.

Depending on the number of transputers to manage within a transputer system there might be several NCUs included. The transputers of all these NCUs form a network where the system software, the *Network Configuration Manager*, is running. The operation of this program will be described in detail in chapter 2.

The main processors of the NCUs are connected in a pipeline. (In a SuperCluster 256 some communication processors are added and a more complex network is built up with the main processors.)

The processor of the first NCU in this pipe is connected via link0 to the *ConfigIn Link* (the system's boot link) of the MultiCluster 2 or SuperCluster. This pipe is even continued beyond a single system via ConfigOut <-> ConfigIn connections to get a MultiCluster 2/64-4 unit (see figure 1.7) or a SuperCluster 128 (see figure 1.13), for instance.

Nine Inmos C004 VLSI-chips on one NCU build up a *matrix crossbar switch* to connect the *communication links* of the transputers belonging to the processing and I/O subsystem available for the users.

Further nine of these chips build up another *matrix crossbar switch* to connect the associated *reset lines* according to the PARSYTEC unique decentralised scheme for system control.

Each matrix crossbar switch has a capacity of 96x96 UniLinks. An intelligent switching strategy allows the propagation delay for a point-to-point connection through the NCUs to be not more than one C004-chip i.e. one level for the SuperCluster 16, the MultiCluster 2/16 and the MultiCluster 2/32. More complex systems such as the MultiCluster 2/64 and larger SuperCluster systems of course need more switching levels.

Levels (maximal number of intermediate C004s for a processor to processor connection):

machine	max number of C004s at one link connection
MultiCluster 2 FT	0
MultiCluster 2/16-1 NCU	1
MultiCluster 2/32-2 NCU	1
MultiCluster 2/64-4 NCU	2
SuperCluster 16	1
SuperCluster 32	3
SuperCluster 48	3
SuperCluster 64	3
SuperCluster 128	4
SuperCluster 256	3*
SuperCluster 320	3*

A third crossbar switch, the *communication crossbar switch*, allows the main NCU processor to communicate to any user or transputer of the system to receive requests.

* A special switching strategy (Clos network) used for the SuperCluster 256 and 320 reduces the propagation delay of a link connection to be not more than that of 3 NCUs !

The main processor controls the crossbar switches and executes the other tasks of the NCM. The sub processor controls the reset level and reports user requests to the main processor by scanning all reset signals led into the NCU. A simple wiring diagram is shown in figure 1.2.

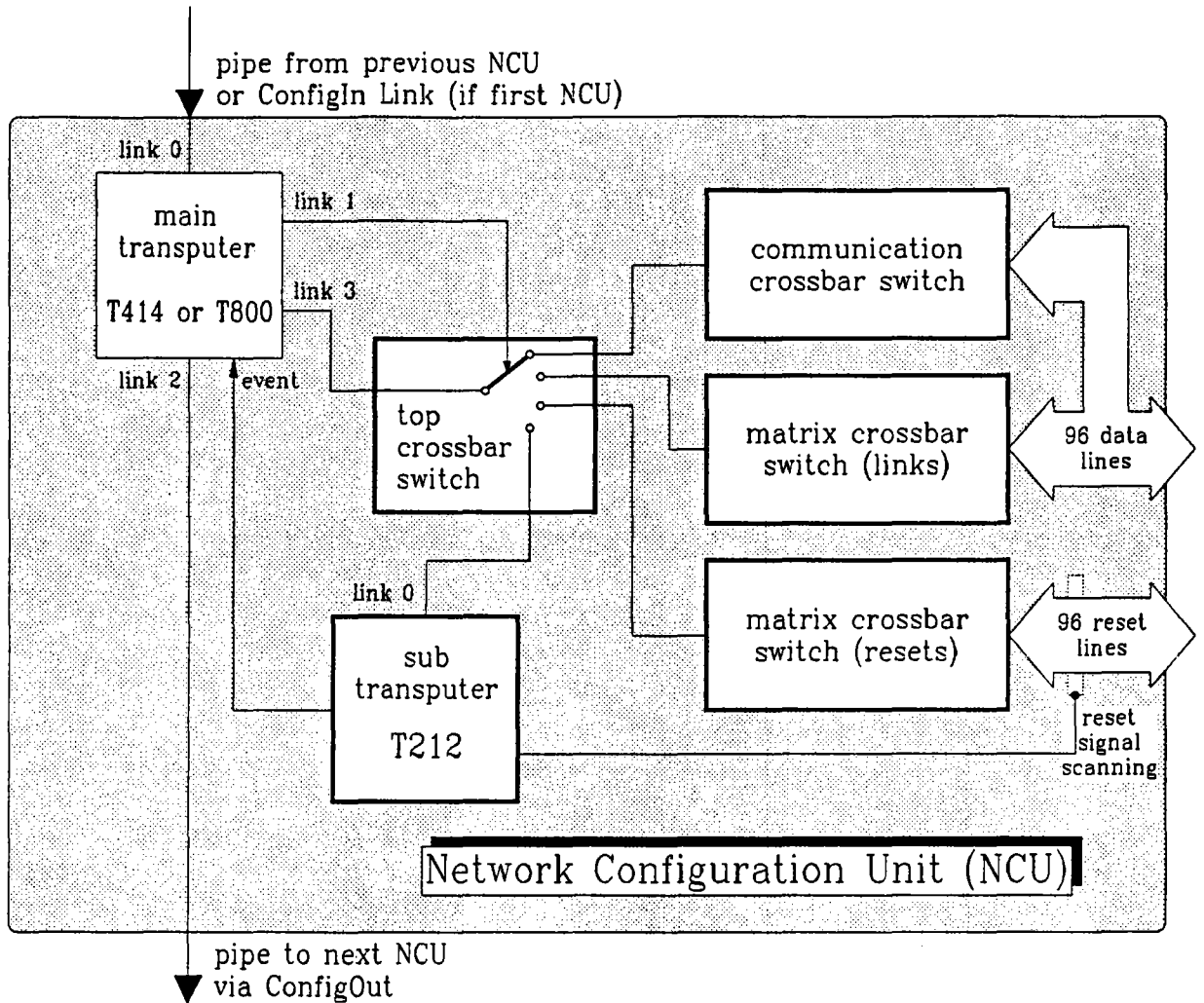


Fig 1.2 The Network Configuration Unit (NCU)

1.2 AVAILABLE TRANSPUTER SYSTEMS

This section will give an outline of the different models' features. They are classified according to the level of reconfiguration and multiuser capability. For each class of systems the user relevant differences are described.

All systems have some common features:

- It is possible to get any number of processors from 1 up to 64 within a MultiCluster 2 and from 16 up to 320 or more within a SuperCluster (special processors not included). Systems with fixed topology are unlimited in their size.
- Special plugs at the rear panel for GDS, TFG, ETN and MSC are available. Detailed information about accessing these plugs is given in the Technical Documentation.
- Up to 8 users can be connected with the multiuser systems (16 users at SuperCluster 256 / 320).
- Upgrade is possible by adding NCUs and processor modules (chapter 1.4).

1.2.1 Fixed Topology

The fixed topology systems (only available for MultiCluster 2) are not reconfigurable and mentioned only for the sake of completeness. The only way to choose a certain network topology is to select one of the available Fixed Topology Modules (FTM).

These FTMs are special boards which are plugged into the unit's configuration slots. This offers the choice of standard topologies to the user such as the matrix or tree structures. Note that another topology is achieved only by altering the FTMs' jumpers. To get electronically reconfigurable systems an upgrade is easy possible (see chapter 1.4).

These systems don't have a configuration control system (i.e. no NCUs) and, therefore, no system software is necessary. Thus, they are not further regarded in this document.

The available machines are:

MC 2/16-FT	MultiCluster 2, 16 processors, Fixed Topology
MC 2/32-FT	MultiCluster 2, 32 processors, Fixed Topology
MC 2/64-FT	MultiCluster 2, 64 processors, Fixed Topology

1.2.2 Logical Reconfigurability

Any processor topology can be switched on machines with logical reconfigurability, although not any link of a processor can be physically connected to any link of another processor.

The MultiCluster 2/32, for example, is not able to connect an even link of one processor to an odd link of another one.

Therefore, the logical link connections of an arbitrary processor topology are mapped to physical link connections which realize the requested topology on the machine.

This linkmapping is done by the interface software, so the user gets the feeling that his original topology has been realised.

The available machines are:

MC 2/32-2 NCU	MultiCluster 2, 32 processors, 2 NCUs
MC 2/64-4 NCU	MultiCluster 2, 64 processors, 4 NCUs

The functionality is a *full logical reconfigurability* and *full multiuser capabilities*.

1.2.3 Physical Reconfigurability

For physically reconfigurable systems any link of any processor or IO module can be connected physically to any link of any other processor or IO module.

No linkmapping is necessary, the processor topology is realised directly on the machine.

The available machines are:

MC 2/16-1 NCU	MultiCluster 2, 16 processors, 1 NCU *
SC 16	SuperCluster, 16 processors, 1 NCU
SC 32	SuperCluster, 32 processors, 2 NCUs
SC 48	SuperCluster, 48 processors, 5 NCUs
SC 64	SuperCluster, 64 processors, 6 NCUs
SC 128	SuperCluster, 128 processors, 12 NCUs
SC 256	SuperCluster, 256 processors, 26-28 NCUs
SC 320	SuperCluster, 320 processors, 30 NCUs

* If the MC 2/16-1 NCU is upgraded, it becomes a logically reconfigurable system.

In all cases the functionality is represented by a *full physical reconfigurability* and *full multiuser capabilities*.

In the following section the various machines of the last two classes will be discussed in detail in order to show the way of connecting and switching the transputers among one another.

1.3 NETWORK RECONFIGURATION

1.3.1 MultiCluster 2/16-1 NCU and SuperCluster 16

For 16 or less processors, a single NCU provides full crossbar function between all links in the transputer network. Each of the 4 transputer links is connected with the NCU. Thus, an arbitrary link switching is possible.

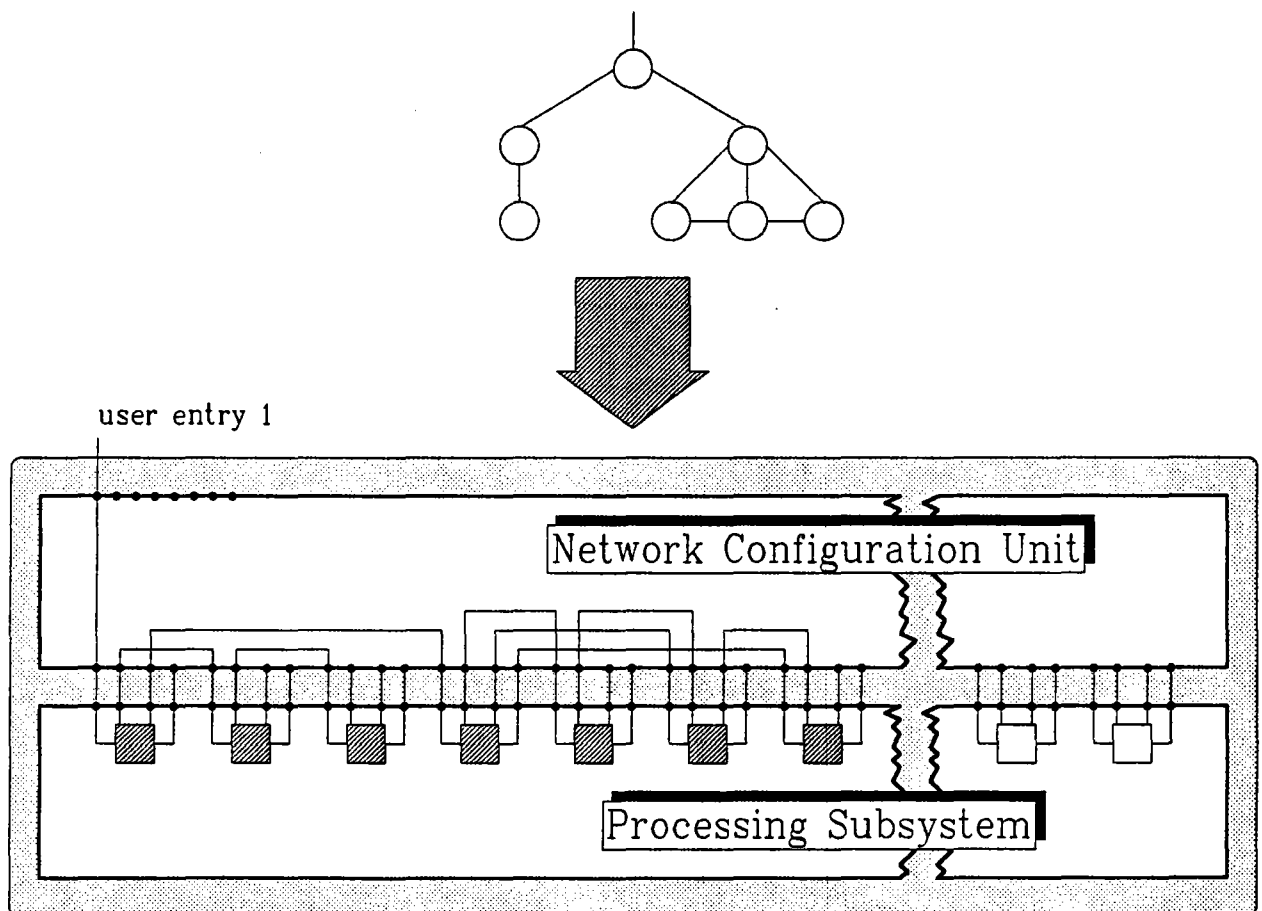


Fig 1.3 Configuration example for MultiCluster 2/16-1 NCU or SuperCluster 16

In order to understand this switching scheme let us think of a user application that needs a certain tree structure. The upper section of figure 1.3 shows this topology. Below you can see how the NCU might switch the transputer links to implement the requested network. Note that only 7 of the 16 processors are involved, and thus, only 28 of the 96 NCU channels are used. The remaining processors and channels may be used for establishing networks of the other users.

How the user sends a request for a network to the MultiCluster 2 will be described in chapter 3. Figure 1.4 shows a wiring diagram of a MC 2/16-1 NCU system and the connected periphery.

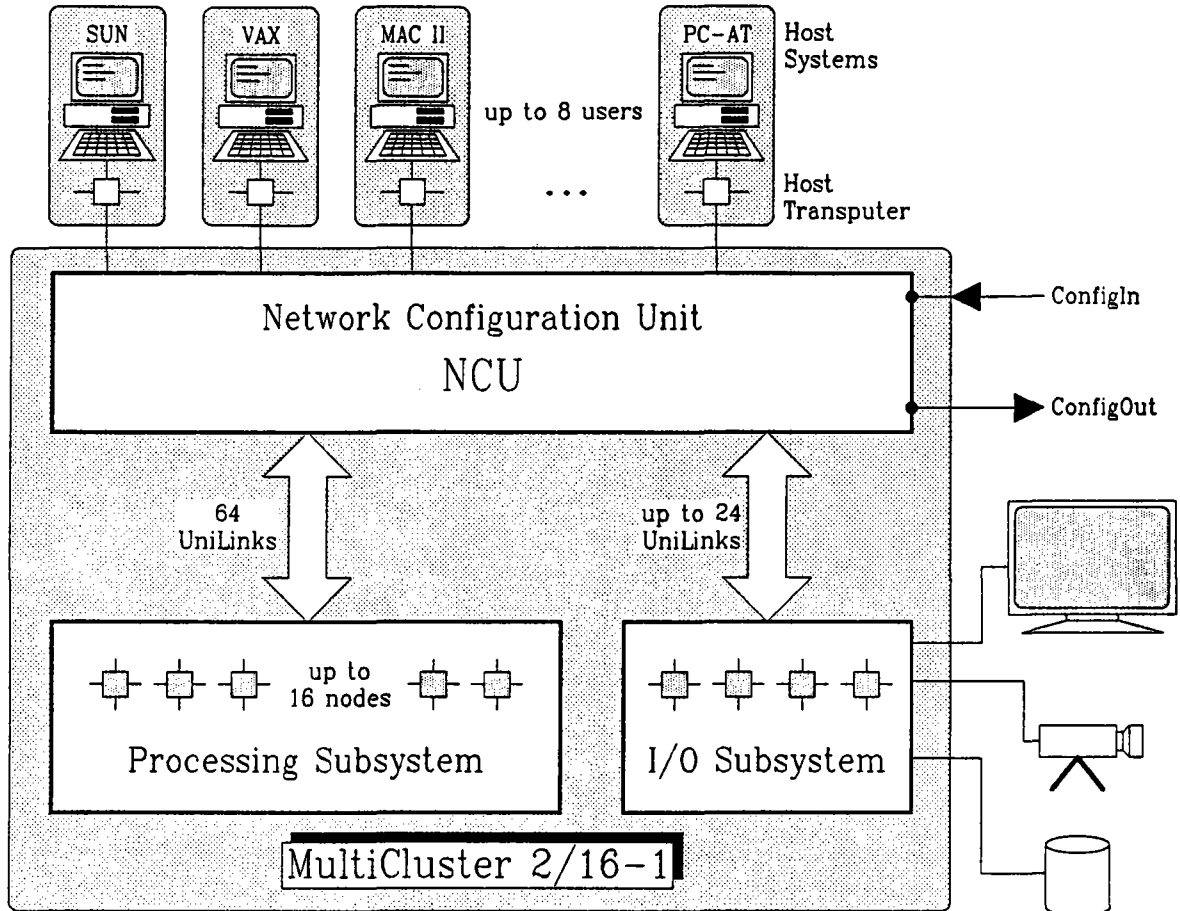


Fig 1.4 MultiCluster 2/16-1 NCU (physical configurability, multiuser)

The SuperCluster 16 differs from the MultiCluster 2/16 in two points:

- Each of the 8 users gets access to the machine via two links (the first and the second user entry).
- The I/O Subsystem has only 16 UniLinks to the NCU.

1.3.2 MultiCluster 2/32-2 NCU and MultiCluster 2/64-4 NCU

For configurations with more than 16 processors on a MultiCluster 2 the links of the processing part (i.e. application and special processors) are divided into an 'even' and an 'odd' group, with a separate NCU for each group.

Thus, each transputer administrated by the system is connected to two different NCUs.

There have to be 2* (rather than 1) NCUs to manage the 32 transputers with their 128 links because a single NCU makes available only 96 channels.

The 2nd NCU extends the number of channels to 192 which is sufficient for the transputer links and which will leave 64 channels for other purposes.

Two approaches are possible:

The first one is to separate the transputers into two groups with 16 transputers each and to connect all their 64 links with one NCU respectively.

The second one (which is used for the MC 2/32 and MC 2/64) is to connect 2 links of each of the 32 transputers with one NCU and the other pair of links with the other NCU.

In both cases 64 channels are occupied on each NCU. But in the first there is no possibility to configure a coherent network with more than 16 transputers. Only the interconnection of the NCUs would remove this lack wasting the remaining channels. But these channels are needed for the special processors and for extending the system.

The latter case will offer a way to realize arbitrary network structures with up to 32 transputers without using the channels for NCU interconnection. Even more than 32 transputers are possible by using the free channels for connections to another 32 processor system (MC 2/64-4 NCU, see figure 1.7).

* To simplify the explanations the numbers given here refer to the MC 2/32-2 NCU. For the MC 2/64-4 NCU all numbers have to be doubled.

Figure 1.5 shows that each NCU gets 64 links of the application processors. The remaining 32 channels on each NCU are used for the special processors (in the same way as for the application processors), for the user entries and for connections between NCUs.

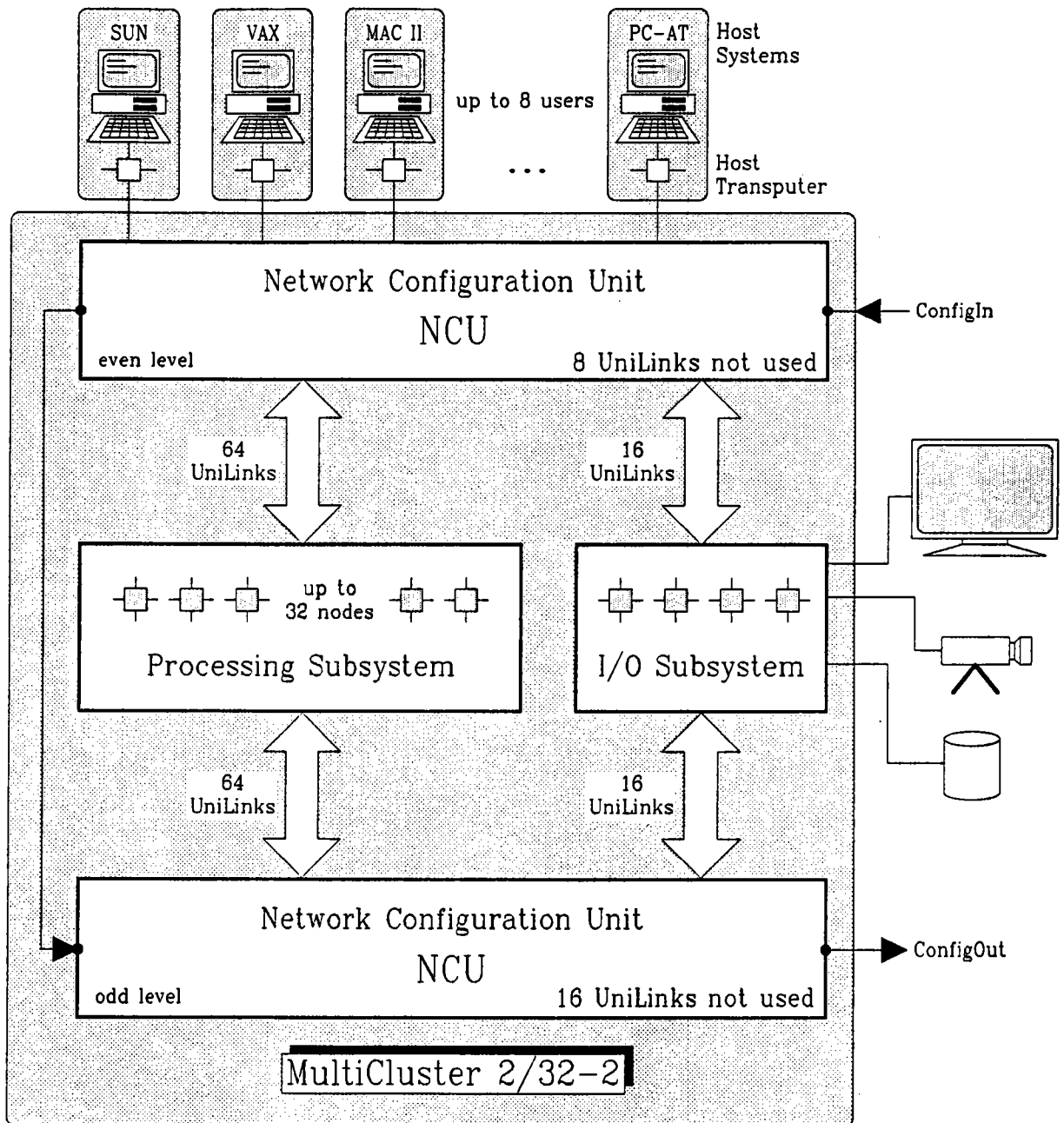


Fig 1.5 MultiCluster 2/32-2 NCU (logical configurability, multiuser)

Does this strategy cause problems concerning link numbering?

To answer that question let us think of an occam PROGRAM or a Helios resource map where link0 of a processor should be connected with link1 of another processor. If we actually want to switch the physical link0 together with the physical link1 this is impossible.

On the other hand each of the 4 processor links are totally equivalent. There is no functional difference between a connection of link0 and link1 or link0 and link2.

Thus, only the structure of the user's link specifications is respected and the numbers are regarded as logical link numbers which have to be mapped to physical link numbers.

You need not worry about those hardware dependencies because the compiler or the Helios command hconfig resp. the NCM will do the link renaming automatically (see chapters 2 and 3).

Figure 1.6 presents an example configuration which could occur if two users request for the networks shown in the upper section. As you can see, each transputer of the processing subsystem has two links switched by the first NCU and two links by the second.

The first NCU gets just the links with even numbers (link0, link2) and the second NCU the links with odd numbers (link1, link3). Thus, the meaning of the names 'even level' and 'odd level' is obvious.

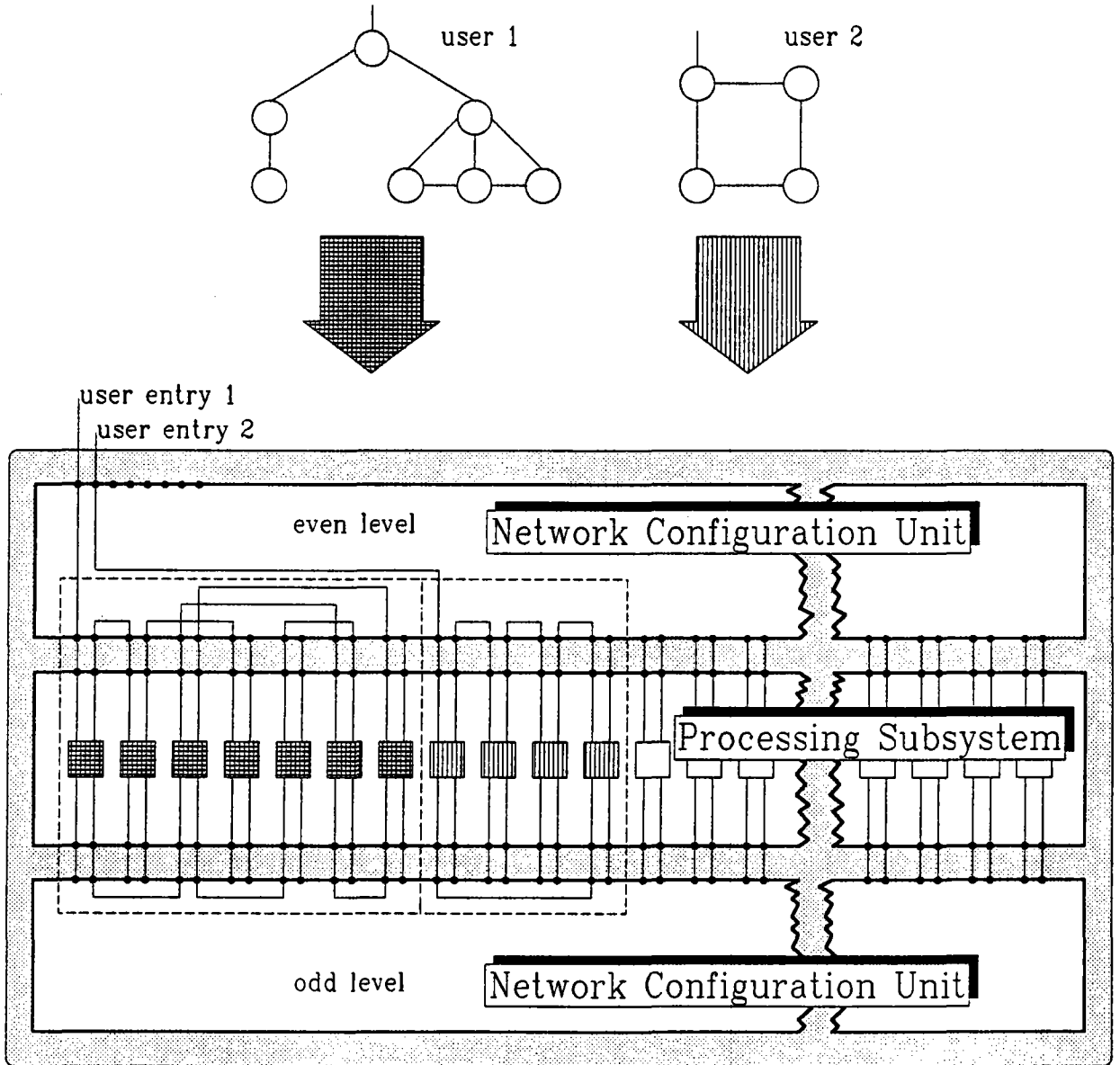


Fig 1.6 MultiCluster 2/32-2 NCU configuration example

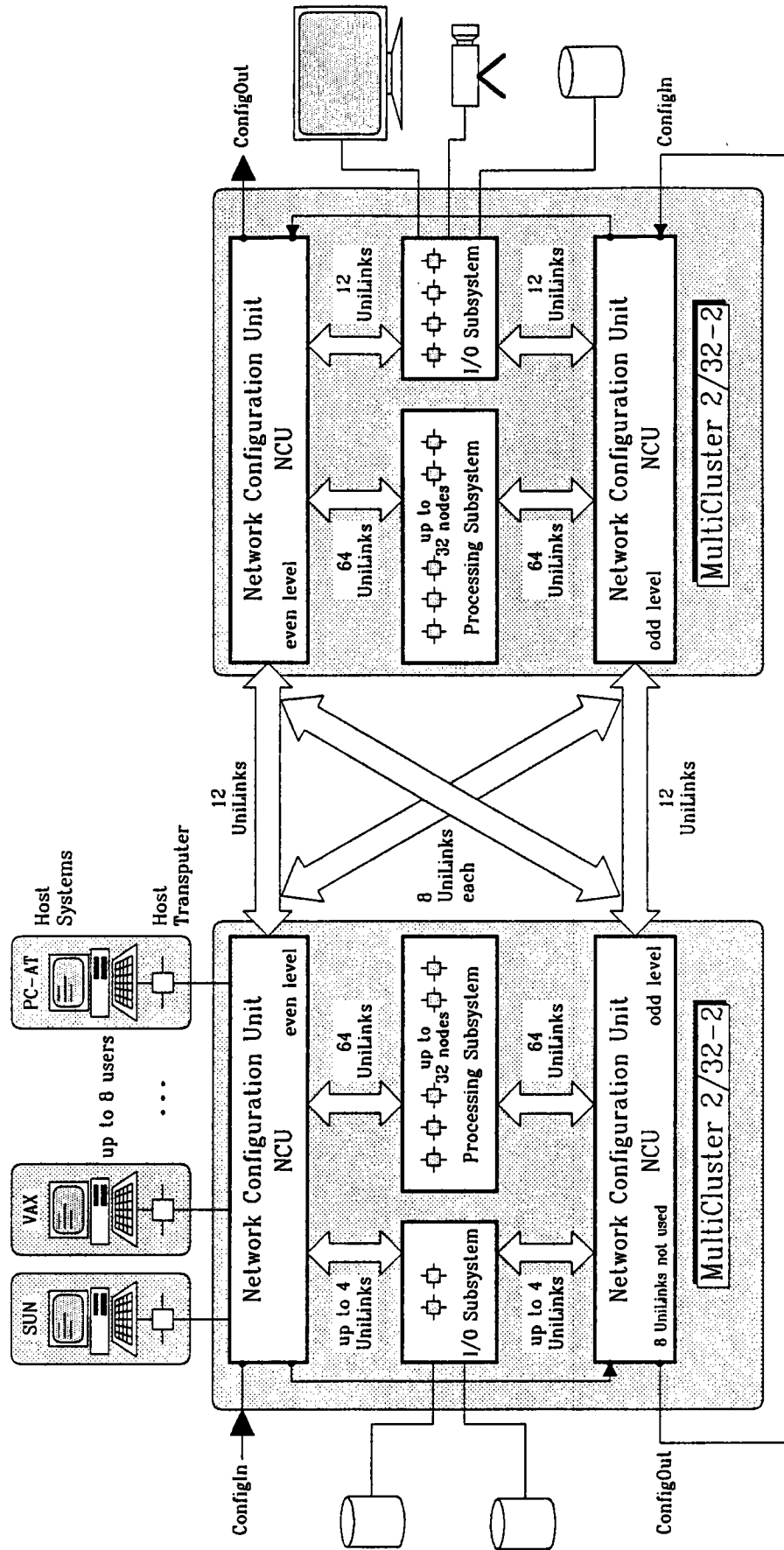


Fig 1.7 MultiCluster 2/64-4 NCU (logical configurability, multiuser)

The MC 2/64-4 NCU is built by connecting two identical MC 2/32-2 NCU systems. To have a fully reconfigurable machine these connections are established between the 4 NCUs.

A total of 80 NCU channels is used to guarantee a highly flexible interconnection scheme. Figure 1.7 shows the wiring diagram of this machine.

The processors of each MC 2/32 unit represent a working cluster. Even the special processors are associated with one of the clusters.

1.3.3 SuperCluster

The SuperCluster is a system with physical reconfigurability. The concept is based on Computing Clusters connectable via a number of Top NCUs at a higher level.

The Computing Cluster

A Computing Cluster is similar to a single reconfigurable machine containing one NCU and 16 processors (cf. MultiCluster 2/16-1 NCU and SuperCluster 16, figure 1.4).

It is used to build up any SuperCluster with more than 16 processors.

16 processors are connected via their 4 links to 64 NCU channels. 32 NCU channels are reserved for inter cluster connections.

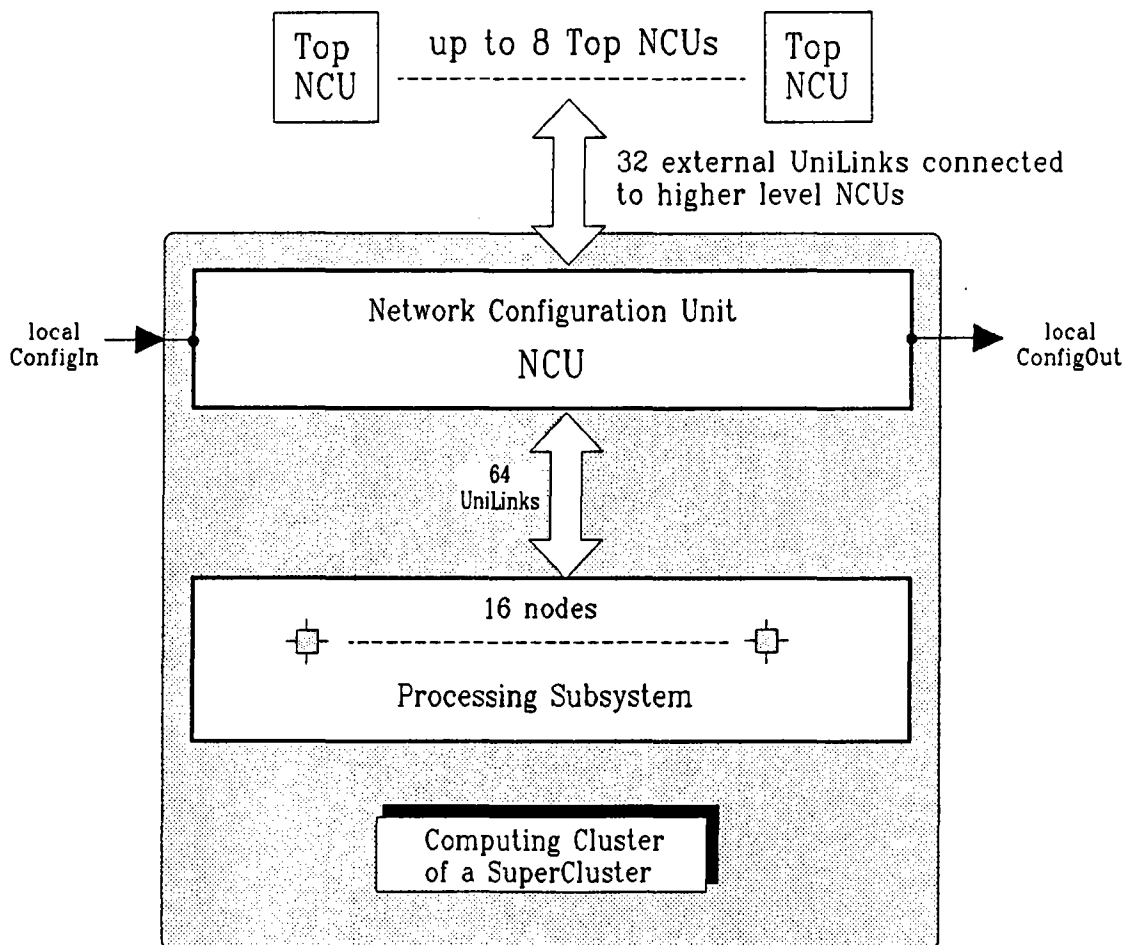


Fig 1.8 Single Computing Cluster (physical configurability, multiuser)

The 16 processors may be connected to processors in other Computing Clusters via 32 external edges. These should be enough to realize any processor topology by partitioning the processors in partial topologies of 16 processors each with less than 32 external edges.

For a user network partitioned into a single Computing Cluster the wiring scheme is similar to that in a 16 processor machine (cf. figure 1.3 "Configuration example for MultiCluster 2/16-1 NCU or SuperCluster 16").

The user entry is led via a Top NCU and an external link into the Computing Cluster.

No I/O subsystem or user entry is connected inside a Computing Cluster. It is a pure processing subsystem with 32 links connected to an upper level.

Connecting Computing Clusters

Suppose that too few processors are free in one Computing Cluster. Then the network is realised using Top NCUs.

The network shown in the upper part of figure 1.9 first of all is partitioned in two parts. Afterwards, the NCM can switch the transputer links as shown below to implement the requested network.

Note that there might be more than one Top NCU which will be used in the same way as shown in figure 1.9 for one Top NCU.

Each Top NCU is connected via 64 of its channels to different Computing Clusters. The remaining 32 channels are used for various functions in the respective SuperCluster. Normally, user entries and the I/O Subsystem are integrated via these channels.

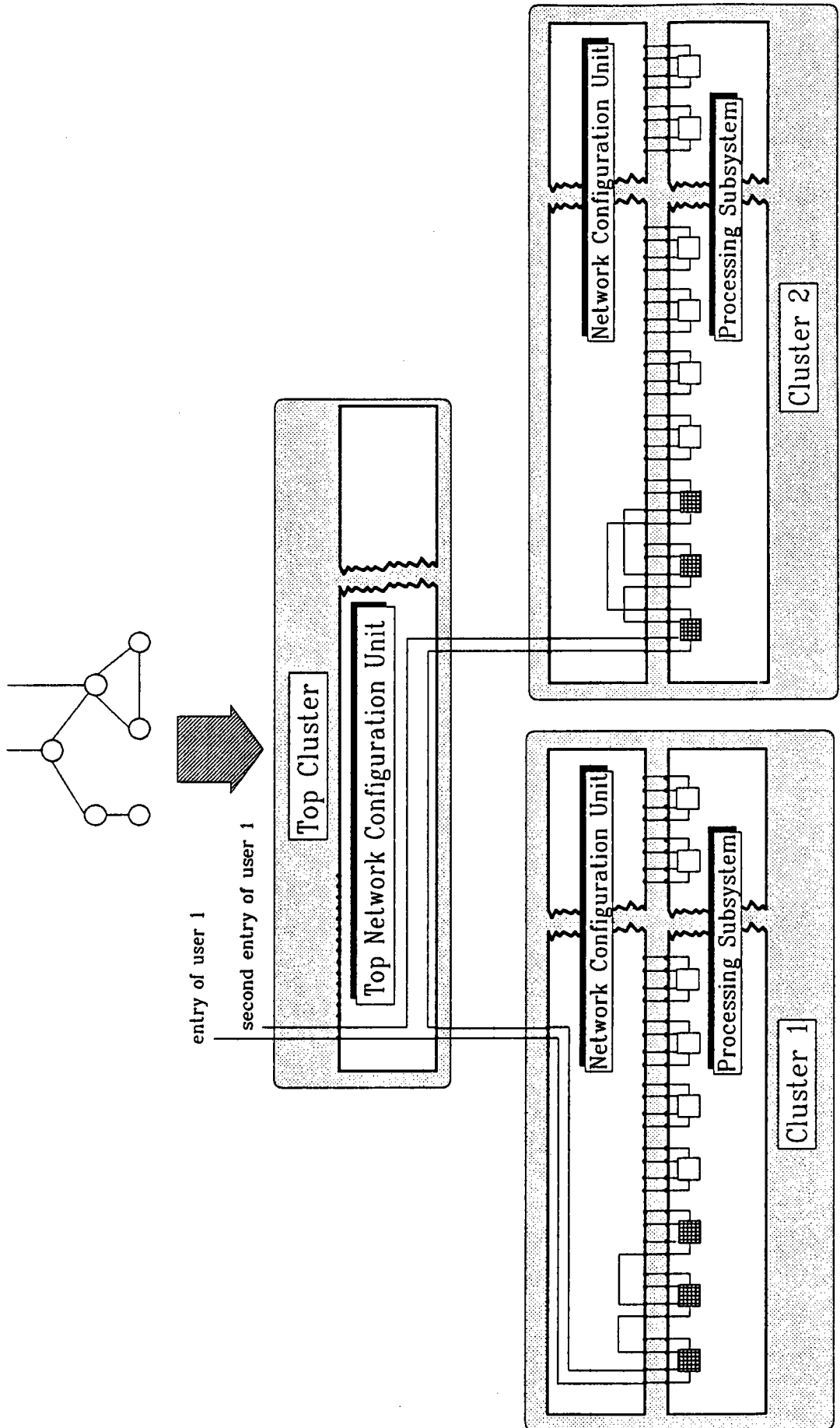


Fig 1.9 Configuration Example via Top NCUs

The following figures show how the various SuperCluster models are built up:

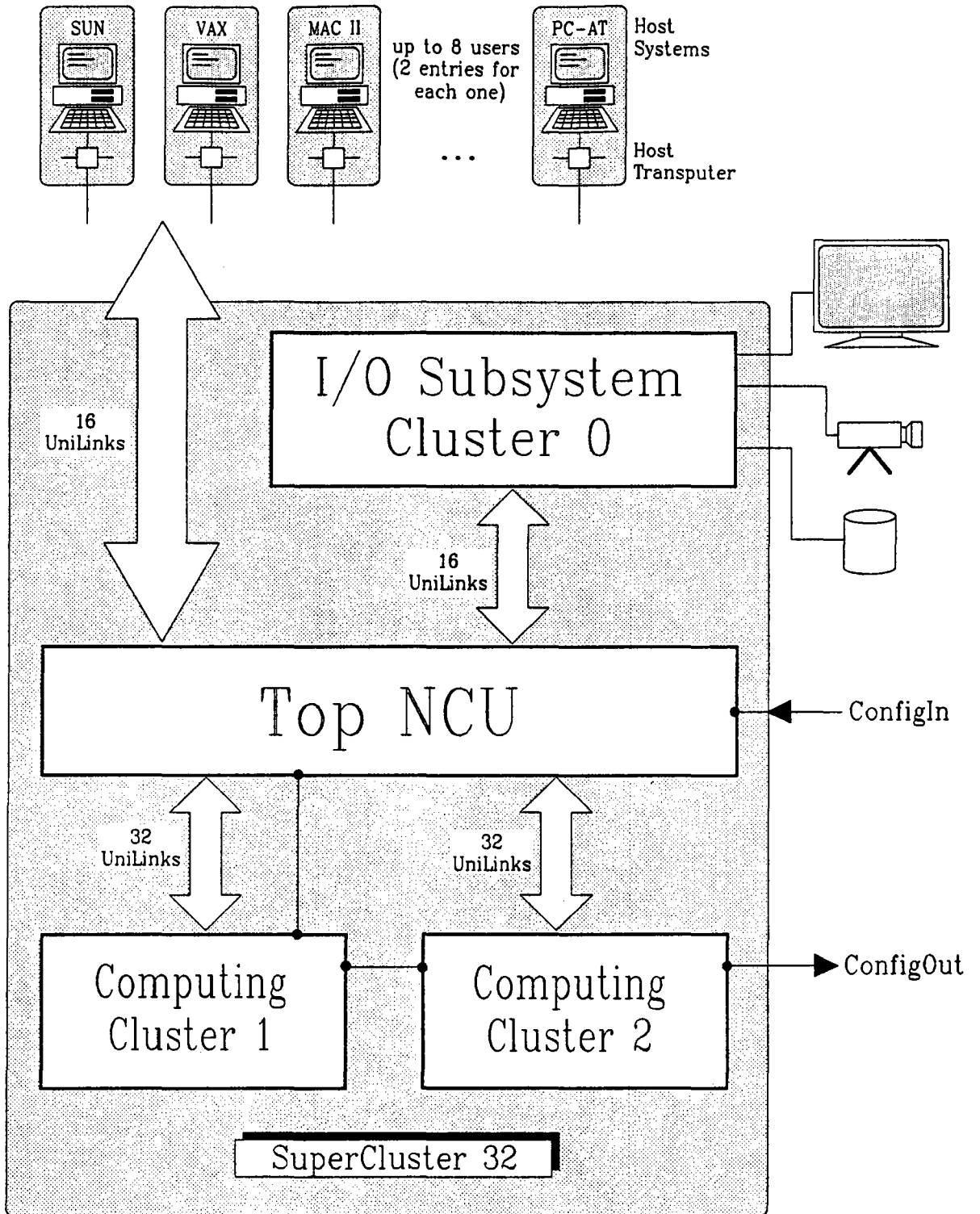


Fig 1.10 SuperCluster 32

The SuperCluster 32 contains two Computing Clusters connected via one Top NCU. The 32 remaining channels of the Top NCU are used to connect 8 users via 2 links each and to integrate up to 4 special processor boards via the I/O subsystem.

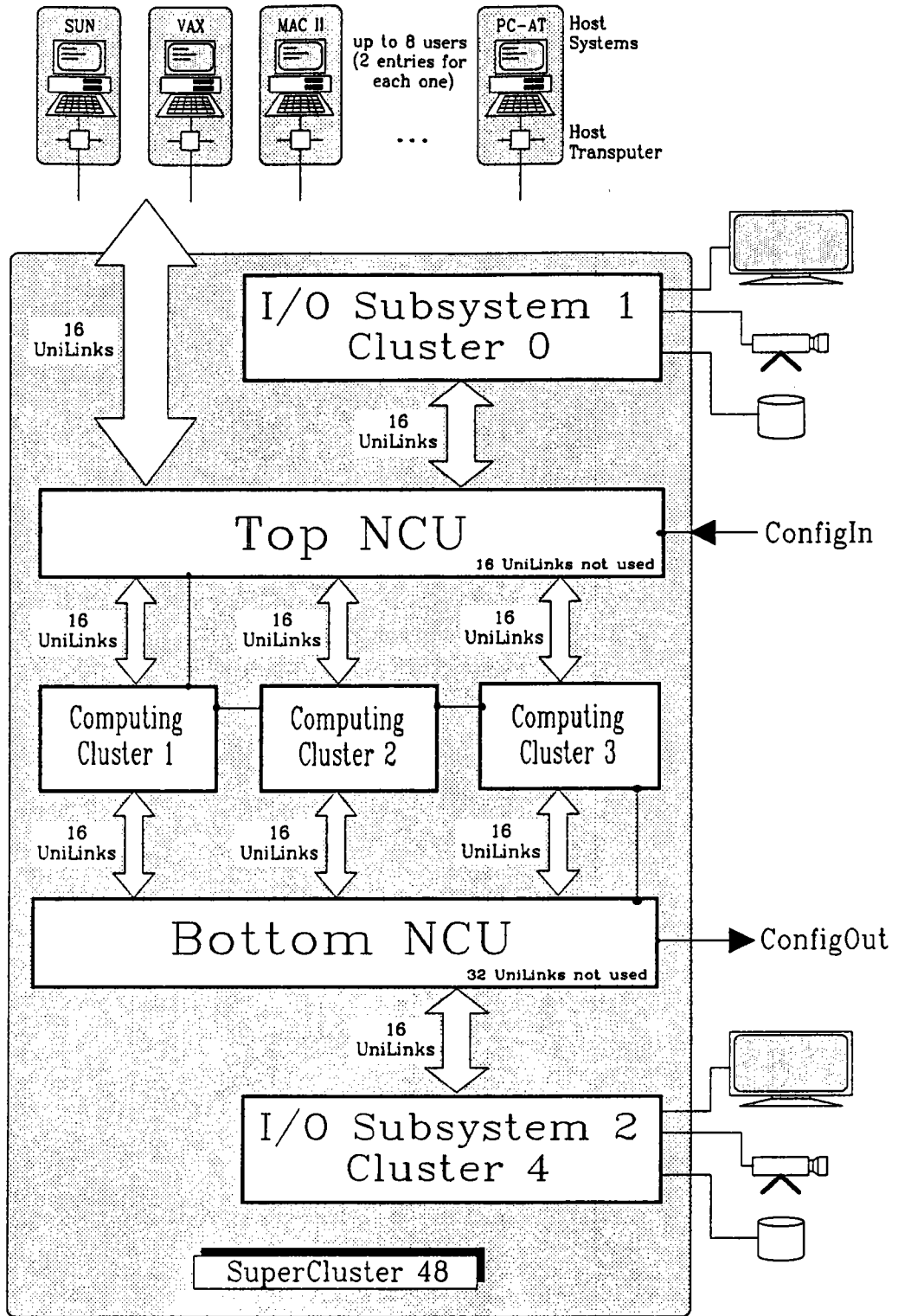


Fig 1.11 SuperCluster 48

The SuperCluster 48 mainly is a SuperCluster 64 with one Computing Cluster removed.

Via the first Top NCU 8 users are connected by two links each. By the 16 remaining channels and 16 channels of the second Top NCU (called Bottom NCU) up to 8 special processor boards are integrated.

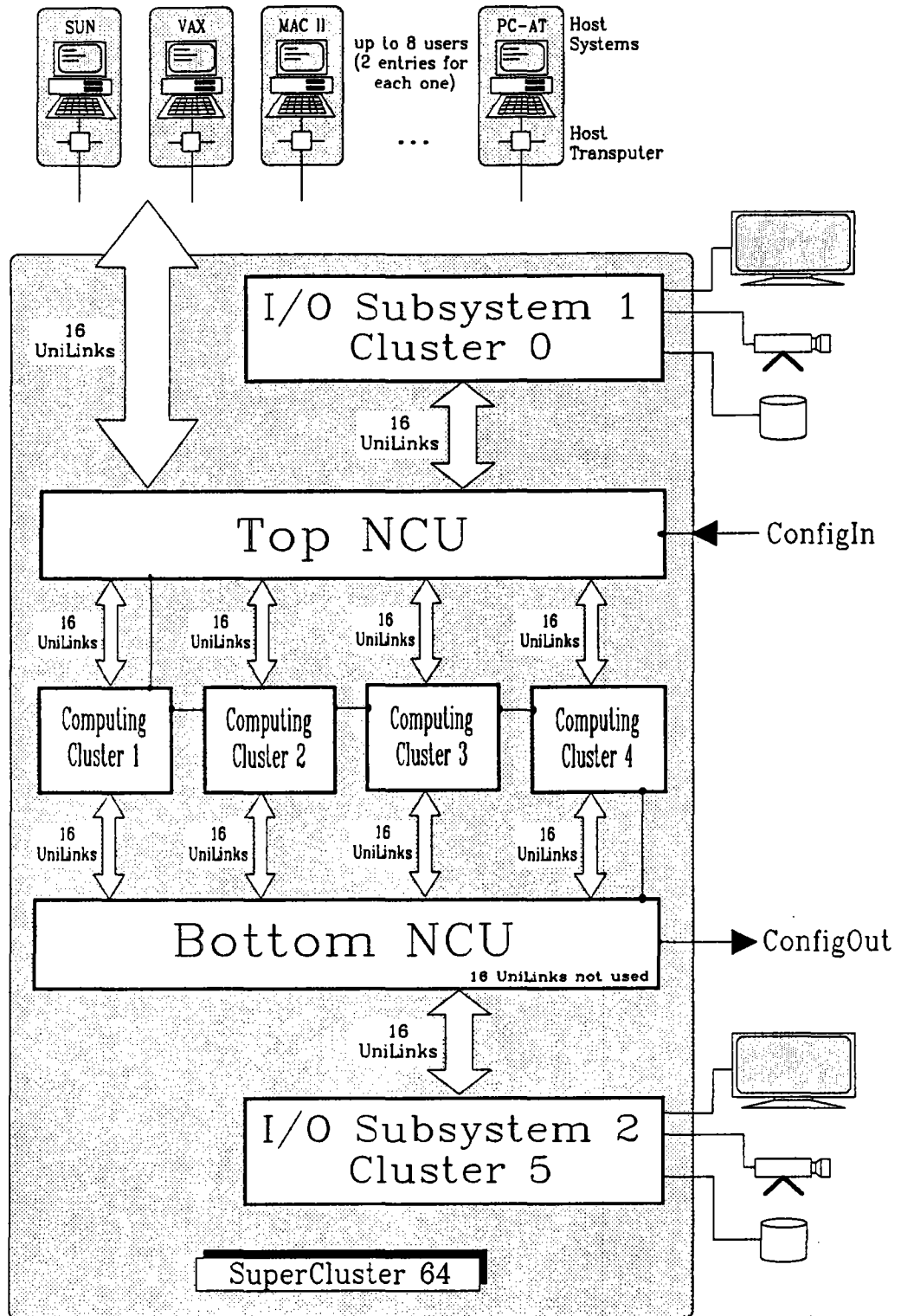


Fig 1.12 SuperCluster 64

The SuperCluster 64 is the basic model of a SuperCluster. A single frame contains 4 Computing Clusters connected via two Top NCUs (named Top and Bottom NCU by their location in the frame).

The I/O subsystem located in the expansion unit of the frame is connected to the Top and Bottom NCU via 16 channels to each. 8 users can be connected with two links each via the (first) Top NCU to the SuperCluster unit.

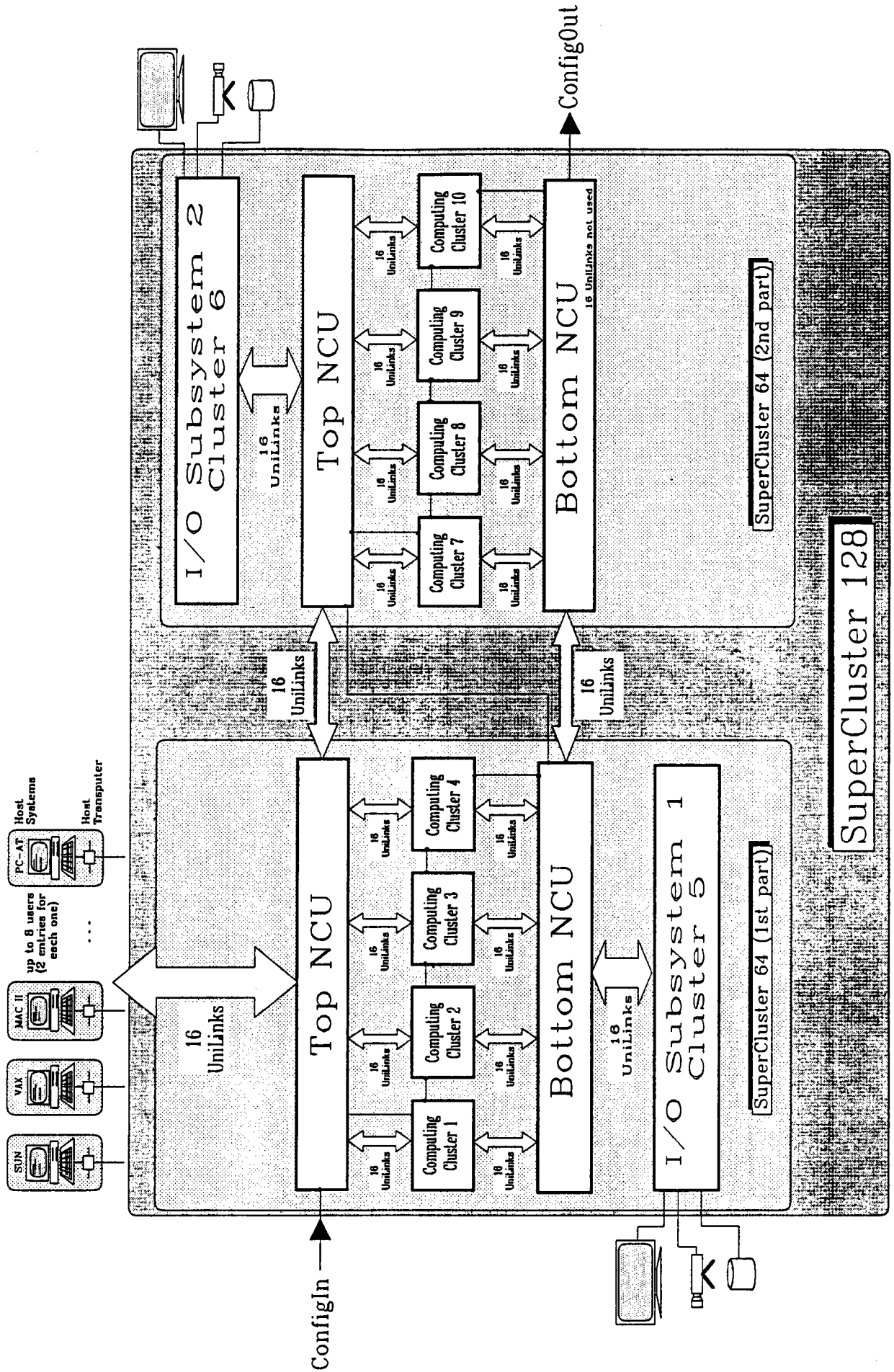


Fig 1.13 SuperCluster 128

Two basic SuperCluster 64 models build up a SuperCluster 128 (see figure 1.13).

The actual NCM for this system supports 32 interconnection channels between the two basic models. Using some extra NCUs the number of interconnections may be increased.

The I/O subsystem has the same size as in a single SuperCluster 64, one part connected to the first and another part connected to the second SuperCluster.

In a similar way a certain number of SuperCluster 64 frames can be connected to more complex systems without changing the hardware:

The SuperClusters 256 and 320 are composed of 16 or 20 Computing Clusters connected via 8 Top NCUs in a Clos network (each of the 16/20 Computing Cluster NCUs is connected to each Top NCU).

Some I/O clusters are built up from one NCU connected by 32 channels to the I/O subsystem and by 64 channels to the 8 Top NCUs of the Clos network.

The user entries also are connected via the I/O cluster NCUs to the system.

Note : In figures 1.9 to 1.13 the cluster numbers are the numbers of the corresponding NCU to which the cluster processors are connected, where the NCUs are numbered according to their position in the NCU pipe (starting with 0).

1.4 SYSTEM UPGRADE AND EXPANDABILITY

The system can always be upgraded with processors up to the given limit. By adding another FTM or NCU the number of processors may be increased furthermore.

Fixed topology may be changed to reconfigurability replacing the FTMs of the machine by NCUs.

1.4.1 MultiCluster 2

In the following figure you see the upgrade possibilities for the MultiCluster 2. The horizontal axis shows the change from fixed topology to configurability and the vertical axis shows the upgrade with processors:

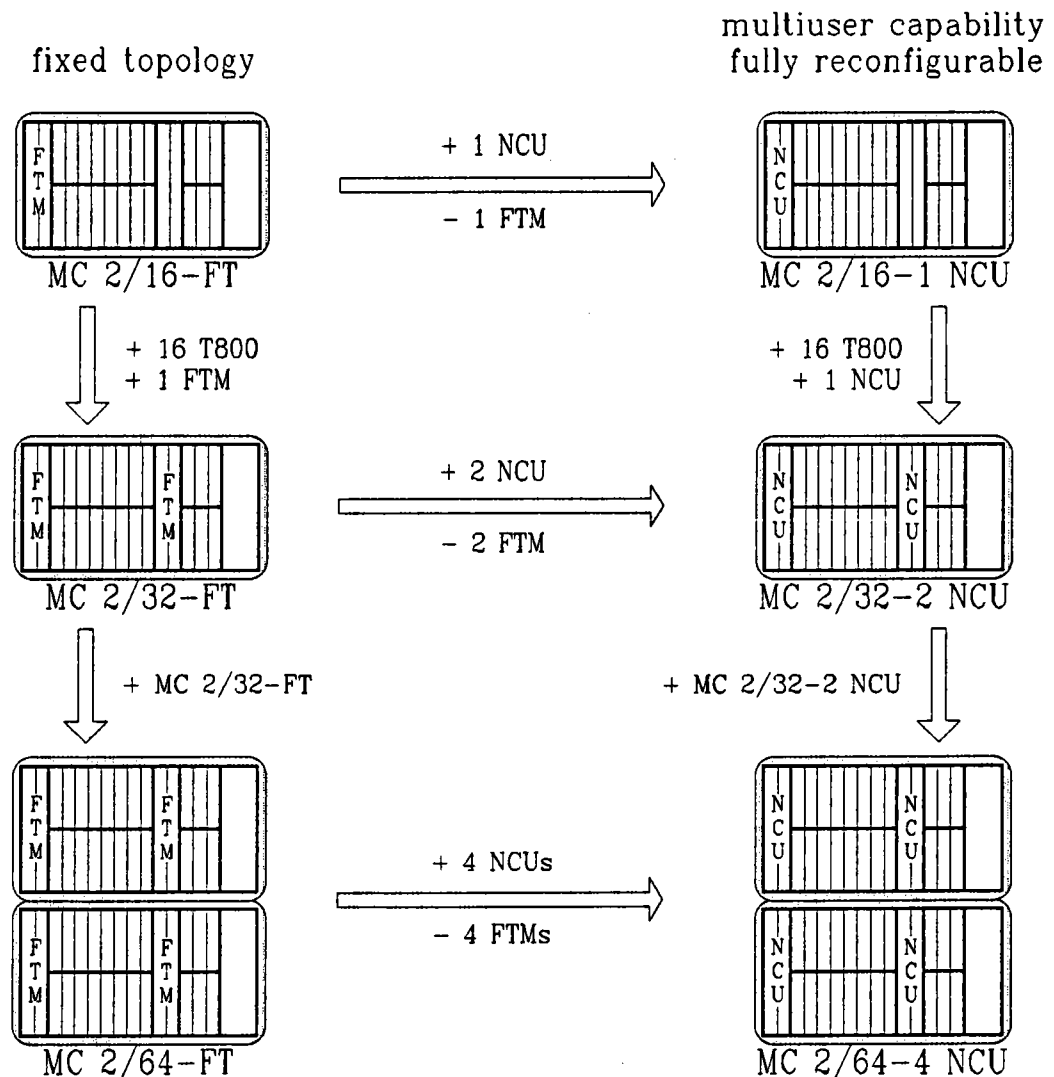


Fig 1.14 Upgrade and Expandability for the MultiCluster 2 Series

1.4.2 SuperCluster

SC 16	1 NCU
+ 2 NCUs + 16 T800	
SC 32	3 NCUs
+ 2 NCUs + 16 T800	
SC 48	5 NCUs
+ 1 NCU + 16 T800	
SC 64	6 NCUs
+ 1 frame + 6 NCUs + 64 T800	
SC 128	12 NCUs
+ 3 frames + 14-16 NCUs (depending on + 2 TPM I/O subsystem) + 128 T800	
SC 256	26-28 NCUs
+ 1 frame + 2-4 NCUs + 64 T800	
SC 320	30 NCUs
.	
.	
.	

Once purchased a SuperCluster frame with a certain number of processors you will always be able to upgrade to any other size by adding NCU and processor boards.

When a frame is fully equipped with the maximum possible number of NCUs and processors, a further upgrade may be done by adding another frame.

There is no fixed limit for the number of processors in one system because adding some more NCUs always gives the possibility to add more Computing Clusters.

2 The System Software

The main task to solve for each new application which should run on a transputer system is to switch the available transputer links in a way that the resulting network will fit the desired application.

The simplest way to achieve this can be found in systems where the user does this task manually with direct access to the transputer links. Due to the complexity of many networks this often is very inconvenient. However, the switching scheme (configuration) exists explicitly and, thus, is visible.

More comfortable is the approach realised in electronically reconfigurable systems. They have own intelligence and full access to the transputer links. Thus, they may switch a desired network according to control information they get from outside.

This is the task of the system software which controls the configuration subsystem, handles user network requests and administrates and configures the resources of the processing and the I/O subsystem.

The multiuser capability of the software allows a full reconfigurability and a real physical partitioning of the transputers into autonomous subnets for every user. This enables a true multiuser operating mode with fault tolerant properties and natural protection among the users, a unique feature.

Each user can individually determine:

- the amount of (free) processing power needed
- the optimal processor topology for his task (configured automatically by the system software)
- the type of frontend workstation to be used
- the software environment desired

In a word the system software manages any network configuration request for all users. Therefore, it is called the *Network Configuration Manager* or *NCM*.

2.1 OPERATION OF THE SYSTEM SOFTWARE

The system software is a program consisting of a number of parallel processes which handle the different resources on one hand and the different user requests on the other.

The resources included in such a system are application processors, special processors, user entries, crossbar switches, etc.

The requests coming from outside may be user requests or dispatcher requests. The user requests are sent via a link connected to one of the user entries (L0 - L7 at the rear panel of the system unit).

For dispatcher requests a special link entry (ConfigIn at the rear panel of the system unit) is reserved.

2.1.1 User Requests

Every time the user asks for a special service of the NCM he has to use some kind of interface program («EXE Config and Boot Network» / extended occam 2 compiler utilities for Multitool and »hconfig« for Helios; for details see chapter 3) which 'knows' how to communicate to the NCM, i.e. how to send a request via the connected link. (The protocol of such a user request will be given at the end of chapter 3 "User Interface Software".)

It is important to obey that the same link also is used as an entry into the user network which may be configured later on. Therefore, the first action an interface program has to take is to generate a kind of event signal which may be detected by the NCM.

This process is called '*ringing*' the NCM. It is the only way to distinguish between usual data streams into the network on that link and special requests to the NCM. Thus, sharing a single link among two processes is possible.

2.1.1.1 Ringing the NCM

The following is a brief description of the ringing process (refer to section 1.1). The exact ringing protocol will be given in section 3.6 "NCM User Interface Protocol".

1. A reset signal of a fixed length is sent via the reset wire (according to the connected link). The subprocessor which is scanning the reset wires will detect it.
2. The sub processor informs the main processor by asserting an event signal.
3. After a short communication to the sub processor, the main processor connects one of its links via the communication crossbar switch to the user link on which the reset was detected.

Now the ringing is finished and the interface program can communicate with the NCM running on the main processor of the NCU according to a well defined protocol.

Any user started request is translated by the interface program into a sequence of commands sent via the user entry link.

The following requests are possible:

- o network configuration requests
- o status information requests
- o returning network request

2.1.1.2 Network Configuration Requests

At this moment only the behaviour of the NCM resulting from a network configuration request is described. How to generate such a request will be presented in chapter 3.

First of all we have to distinguish between two general types of systems which handle a configuration request in different ways. The basic hardware differences between these groups and the specific problems concerning link renumbering have been presented in section 1.3.

- **MultiCluster 2/16-1 NCU and SuperCluster**

During a network configuration request the NCM gets the whole information about the network structure the user needs for his application. This includes the logical processor numbers and their interconnections via the assigned links (cf. figure 2.1 "Network configuration by the NCM", first step).

The NCM analyses this information and tries to allocate the necessary resources. In case of success the resources are marked as assigned to the specific user. If the requested resources are not available or other problems occur, an appropriate message is sent back to the interface software.

For clustered machines the NCM partitions the user networks in a way which minimizes intercluster connections (figure 2.1, second step).

The next step is to program the crossbar switches to connect the links of the associated processors according to the network structure. Done this a network is at the users disposal which does not differ from a manually connected one.

All information concerning user networks is held by the user administration process as part of the NCM. It will guarantee that the user keeps his network as long as he needs it. Furthermore all used resources will be locked for other users.

The distribution of resources to users is not priority dependent and so the users are served in a first in/first out way each at a time. Thus, no conflicts may occur between the different requests.

Note that for this group of machines no special treatment of the link numbering is needed.

- **MultiCluster 2/32-2 NCUs and MultiCluster 2/64-4 NCUs**

In contrast to the previous group of machines this group needs to process a link renumbering due to the special hardware characteristics of these systems (cf. figure 2.1, second step).

The basic operation of the NCM does not differ from that in the systems described above with the exception of the crossbar switches programming.

When the NCM receives the description of the network to realise, it will check the link connections.

Usually, the given link numbering will not meet the hardware requirements. Therefore, a link renumbering will take place which does not change the structure of the network. If this process succeeds, the modified network will be switched.

The information about the modified network is sent back to the interface software which has to take care for the new configuration. In which way the particular interface software will handle this situation will be described in chapter 3.

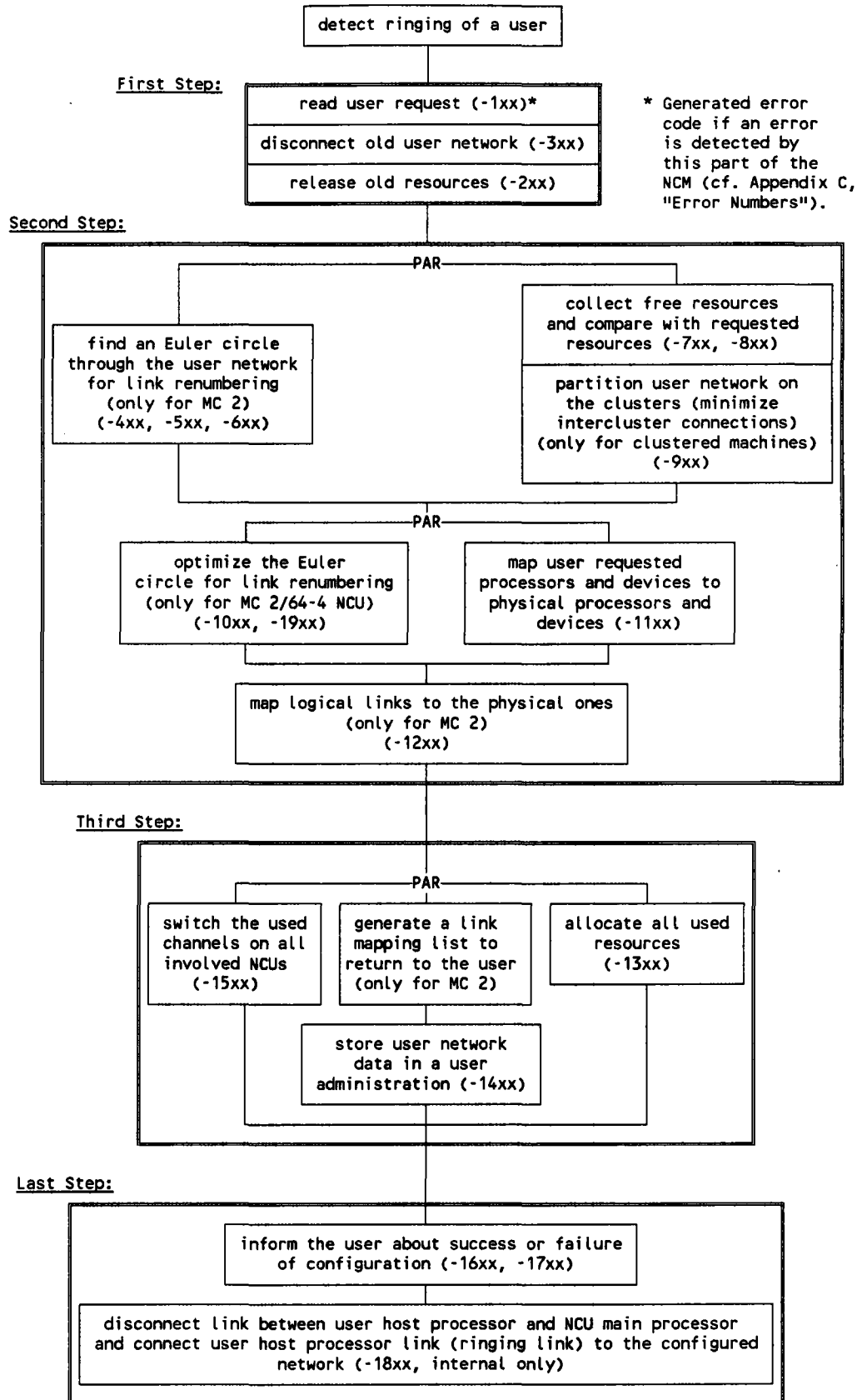


Fig 2.1 Network configuration by the NCM.

2.1.2 Dispatcher Requests

The task of the dispatcher is to load the system software and to proceed other privileged operations such as testing and asking for status reports.

All data are passed through the ConfigIn link which is directly connected to the main processor of the first NCU. A special process of the NCM will listen on this channel for any request.

The main task, however, is to load the system software after power up. Afterwards, the system is ready to accept requests via this link.

The existing interface software packages which will generate these requests and process the booting are »EXE NCM.control«, used with MultiTool, and the Helios program »boot_ncm«. The usage of these tools will be described in the next section.

Requests may also be generated by other software driving the ConfigIn link protocol given in the next section.

Obviously the ConfigIn link is a bit sensitive due to the fact that loading the system software a second time will cancel all existing configured networks (the system software resets all crossbar switches during start-up !).

If an arbitrary program is accidentally loaded via this link, the system software will of course be removed from the NCU processors. This behaviour of the system is not unexpected since the configuration control system is a common network which allows to load transputer programs.

But usually, the settings of the crossbar switches responsible for the user networks are not changed and, thus, the configured networks will remain. Before the system software is loaded once more all users should be given time to finish their work.

2.2 RUNNING AND USING THE SYSTEM SOFTWARE

Due to the fact that the direct access to the system software allows privileged operations the procedures described in this chapter should only be made by the dispatcher.

For example every booting of the system software will cancel all existing networks. It has the same effect as booting a usual workstation.

Booting and using the system software is done via a defined protocol at the ConfigIn link.

Existing tools are the MultiTool »EXE NCM.control« and the Helios program »boot_ncm«.

The system software once booted by any tool (e.g. from MultiTool) allows the access from any other tool (e.g. Helios »boot_ncm«) driving the right ConfigIn link protocol.

For the system software interface there are two modes for protocols, an interactive and a non-interactive mode.

The menus in the interactive mode are generated by the NCM itself (cf. »EXE NCM.control«), in the non-interactive mode the protocol has to be interpreted by the interface software (cf. »boot_ncm«). This type of interface software may be used for batch files.

To make the interactive tool »NCM.control« available for Helios users as well, a restricted nucleus of MultiTool is delivered together with the system software for MultiTool.

Boot Procedure

Booting the system software means to load the code of the Network Configuration Manager, which is held on a so called bootstation, onto the configuration subsystem of the MultiCluster 2 / SuperCluster, where the processors of the NCU(s) will run the NCM.

The Network Configuration Manager can be loaded from any host machine which is able to boot a transputer via a link and drive a communication afterwards.

The first step to take is to connect the bootstation with the MultiCluster 2 / SuperCluster. A link (load link) of the host machine has to be connected to the ConfigIn link of the MultiCluster 2 or SuperCluster (see Technical Documentation and figure 1.1).

This will establish a line between the host machine and the main processor of the first NCU (cf. figure 1.2).

The further procedure depends on whether you work with MultiTool or with Helios and will be explained in sections 2.2.1 and 2.2.2 respectively. Section 2.2.3 explains how to write your own interface software.

Hint : The System Software is delivered in two different types, one for NCUs with T414 as main processor (systems delivered till approximately November 1990) and one for NCUs with a T800 as main processor.

Normally, the two NCU types are not mixed in one system. In case of a mixed system choose the NCU 0 (NCU connected to ConfigIn) to be a T414 NCU and then run T414 software.

2.2.1 MultiTool »NCM.control«

2.2.1.1 Booting

The MultiTool interactive »EXE NCM.control«, the system interface software to load the NCM via MultiTool, is located in the following fold (MultiCluster 2 case, replace 'MultiCluster 2' by 'SuperCluster' and remove '-y' for SuperClusters):

```

{ { { NCM.TOP
{ { { ncmxx.tsr

Hint: Be careful using the EXE NCM.control while users are running
---- their programs. Some functions have access to user resources.

Network Configuration Manager ( NCM ) :

...F CODE EXE NCM.control v 3.0 [901204]
... NCM Software MultiCluster 2/xx-y v 3.0 [901204]

...F How to load the Network Configuration Manager :

}}
}}
```

To meet the requirements of the various hardware configurations different types of the system software are available. Which type you have is indicated by 'MultiCluster 2' or 'SuperCluster' respectively and the values of the xx-y extensions. xx stands for the size of the MultiCluster 2 / SuperCluster and y (only in case of MultiCluster 2) for the number of NCUs in the system.

Entered the above fold we are ready to transfer the code for the NCU(s) which is contained in three folds named »NCM.init«, »T4 NCM« (for T414 NCUs) and »T8 NCM« (for T800 NCUs) collected within the fold »NCM Software MultiCluster xx-y v 3.0« or »NCM Software SuperCluster xx v 3.0« respectively.

The utility program »EXE NCM.control« will help to simplify the booting procedure. First it takes the fold »NCM.init« and sends it to the main processor(s) of the NCU(s) using a special protocol.

There the NCM.init program switches the main processor link(s) onto the sub processor link(s). Thus, the system software for the sub processor(s) can be loaded. In a second step the NCM itself is loaded onto the configuration subsystem.

Earlier version of `NCM.control` load the first fold named `»NCM«` (resp. `»Tx NCM«`).

From version 3.0 on the `NCM.init` program detects the type of the NCU processor network. Using the `NCM.control` version 3.0 the NCM code is loaded in dependence of the NCU main processors' type.

In case of a T414 network the fold `»T4 NCM«` is loaded. If this fails, `NCM.control` loads the first fold named `»NCM«` (resp. `»Tx NCM«`). In case of a T800 network an analogous procedure is done.

In case of a mixed network or if `NCM.init` does not detect the NCU main processor's type (old `NCM.init` version), the `NCM.control` loads the first fold named `»NCM«` (resp. `»Tx NCM«`) and if this fails, it tries to load `»T4 NCM«`.

Summary of the Necessary Steps

The following list shows how to boot the MultiCluster 2 / SuperCluster. Note that the booting procedure is task of the dispatcher and has to be done only once after power on.

Every command of the NCM menus is executed only by pressing a single keystroke which is enclosed in `< >`.

The MultiTool functions are typed enclosed in `[]`. To find out which keys are used on your specific host machine to execute them (as `[ENTER FOLD]`, for instance) please press `»ESC h e«` or function key `»F1«`.

```
NCM.control (vers. 3.0): System Interface to the Network Configuration Manager
                        Copyright (C) 1990, Parsytec GmbH

<L> Load Network Configuration Manager (via link 3)
<M> Start NCM monitor
<S> Setup load link
<Q> Quit

:
:

Enter command:
```

Fig 2.2 Start menu of EXE `NCM.control`

1. Switch on the MultiCluster 2 / SuperCluster.
2. Connect a workstation / hostsystem (via its load link) with the MultiCluster 2 / SuperCluster (ConfigIn link) as bootstation (for the dispatcher).
3. Start MultiTool by typing »mtool« or »ncm« (and a blank) at the directory where the system software is located.
4. When »... F NCM.TOP« appears, enter the system software fold by pressing the [ENTER FOLD] key two times.
5. Get the »EXE NCM.control«:
Place the cursor on the fold »CODE EXE NCM.control« and press the key [GET CODE].
6. Place the cursor on the fold »NCM Software« and run the loaded EXE by pressing [RUN EXE]. The menu shown in figure 2.2 will appear.
7. Type <s> (Setup) for selecting the load link (connected to the ConfigIn plug of the MultiCluster 2 / SuperCluster).
8. Type <l> (Load Network Configuration Manager) for loading the NCM. If the loading succeeds, the text below will be printed on screen:

```
Network loaded OK
Awaiting message from: NCM.init startup ok
NCU processor is Txxx
```

```
Network loaded OK
Awaiting message from: Tx NCM startup ok
```

On SuperClusters you can perform a processor check after loading. Bad or not found processors are allocated afterwards.

When loading is finished, the NCM monitor will be reached (cf. figure 2.3) which is described in detail in section 2.2.1.3. Error and warning messages which may occur will be explained in the section below. The functions of the NCM monitor should not be executed while users are running their applications as long as the dispatcher does not know the consequences. In detail <n>, <c>, <d> or <r>, cf. section 2.2.1.3, can disturb network configurations.

9. If the MultiCluster 2 is not fully equipped with the possible number of processors (e.g. 14 processors in an MC 2/16-1 NCU), you should allocate the free slots by the option <a> of the NCM monitor (see section 2.2.1.3 'Allocate free processors').

Special processors should be allocated by option (also for SuperClusters).

10. Type <q> (Quit) to leave the NCM monitor and to return to MultiTool.
11. Type [EXIT FOLD] two times and [FINISH] to leave MultiTool.
12. Disconnect the link cable from the ConfigIn plug of the MultiCluster 2 / SuperCluster. This will prevent you from loading other programs via this link accidentally.

Now the MultiCluster 2 / SuperCluster is ready to be used for every connected host system.

WARNING

Reloading and rebooting of the NCM software will undo all existing network configurations and all user programs. Therefore, be sure that no user is running a program on the MultiCluster 2 / SuperCluster when booting and loading the NCM software.

2.2.1.2 Error Situations and Warnings

When loading the system software as described above, the following warnings or error messages may appear at step 8.

1. WARNINGS

If the NCM is already running on the NCU, a second loading would destroy all existing user networks. Thus, the following message is printed on the screen:

```
The Network Configuration Manager Version 3.0 is
running !
Rebooting resets all user networks !
Do you really want to load ? (Y/[N]) -
```

If you answer yes by pressing y, the NCM will be loaded again and all configurations will be reset. If you answer no by pressing n, no action is taken and the startup menu is entered again.

If NCM.init detects a NCU network mixed of T414 and T800, the following warning is displayed:

```
Network loaded OK
Awaiting message from: NCM.init startup ok.
Warning : The system contains a mixture of
          T4 and T8 NCUs !
```

The T800 code never runs on T414. If T414 code is used, it depends on the type of the NCU 0 (administration process NCU) whether the NCM will work or not.

Choose the NCU 0 to be a T414 NCU and use T414 code to run the NCM.

2. ERRORS

If you get the following error message ...

```
Loading: NCM.init ...
Communication error : Failed to output boot code
                      for processor 0
```

```
Failed to load: NCM.init
```

... there is no way for the host processor to reach the communication subsystem. Thus, the boot code of the NCM can not be sent to the NCU. Several reasons are possible:

- o The specified boot link is wrong. You should change the boot link by entering the startup menu and by choosing the <s> option.

- o There is no physical connection (no wire or a broken wire) between the host processor link and the ConfigIn link of the MultiCluster 2 / SuperCluster.
- o Other reasons which result in a never responding NCM (no power, no NCU boards inserted, etc.).
- o Same message for »T8 NCM«: You may have some NCUs with T414 main processor in your system.
- o Perhaps the MultiCluster 2 / SuperCluster is not switched on.

If you get the following message ...

```
Network loaded OK
Awaiting message from: NCM.init
```

```
Failed to switch on link level of NCU x.
```

... a C004 may have got a protocol error or the power up reset has failed.

Power off the MultiCluster 2 / SuperCluster, wait for some seconds and power on again.

The same message may occur for the reset level.

The following message is displayed if the NCU 0 (administration process NCU) has the wrong type (T414 code loaded on a T800 processor):

```
Network loaded OK
Awaiting message from: NCM.init startup ok.
Warning : The system contains a mixture of
          T4 and T8 NCUs !
```

```
Network loaded OK
Awaiting message from: NCM          : Wrong processor type
                                     on first NCU.
T4 NCM code on T800 is not possible : Please place
T8 code inside the fold and name it by "T8 NCM".
For mixed systems please choose a T414 NCU as NCU 0
and use "T4 NCM" code.
```

The administration process uses floating point operations which are not compatible for T414 and T800. So the right software has to be available in the fold »NCM Software ...«.

For mixed systems there may be a way to run the T414 NCM. If you choose the NCU 0 to be equipped with a T414 main processor, the NCM may work with other NCUs having a T800.

2.2.13 The NCM Monitor

To use the NCM monitor the EXE `NCM.control` has to be used serving as an interface between the dispatcher and the NCM (the host system has to be connected with the MultiCluster 2 / SuperCluster ConfigIn link).

Keyboard input is led through the monitor to the NCM and NCM output is led through the monitor to screen. All menus are coming directly from the NCM which is running in the interactive mode for the system interface.

There are two ways to reach the NCM monitor.

The first one is to follow the steps presented in the previous section. After successfully loading the NCM, you will automatically enter the monitor.

The second one can be used if the NCM is already running: start the EXE `NCM.control` and type `<m>` in the appearing startup menu (setup the right load link to ConfigIn before).

```

      NETWORK - CONFIGURATION - MANAGER
                    Version: 3.0
Commands:
C:      Connect channels
D:      Disconnect channels
S:      Show NCU status
P:      Check free processors
R:      Reset all NCUs
N:      Return a user network
U:      Display processor usage
V:      Display usage of administrated devices
X:      Display channel usage
A:      Allocate free processors
F:      Deallocate processors
Z:      Allocate all free processors
Y:      Release all processors
B:      Allocate free administrated devices
G:      Deallocate administrated devices
O:      Omit random behaviour for partitioning
Q:      Quit

Enter command:

```

Fig 2.3 Menu of the NCM monitor

With the commands of the NCM monitor the dispatcher may directly change the state of the MultiCluster 2 / SuperCluster. You should have access to the Technical Documentation to look up several physical indices.

There are three main fields of commands:

- o Direct hardware access commands

As mentioned in chapter 1 each NCU has a 96x96 crossbar switch (cf. figure 1.2). The settings of these switches may be manipulated for testing purposes. To get an overview of the switching scheme the <S> command is useful.

- o Device administration commands

If a system is not equipped with the maximum number of (special) processors, these 'missing' devices have to be allocated. Thus, the NCM does not take these nodes into consideration while distributing the resources.

- o NCM control commands

The random behaviour of the partitioning algorithm can be disabled for testing hardware or to get an optimal partition for a special network topology. Also user networks may be returned by the dispatcher.

The possible options are described below.

Note for all functions:

The user cannot configure a network while the dispatcher calls one of the following system functions until the dispatcher returns to the main menu (cf. figure 2.3).

1. <C> Connect channels

Purpose : This command allows the 'manual' connection of NCU channels for test purposes. The specified channels will be connected directly on reset and link level.

If channels are used, the corresponding channel connection will not be established.*

Usage : First the NCU has to be selected on which the switching should take place. Therefore, the physical NCU number (cf. Appendix D "Physical Device Numbers") has to be entered (-1 will terminate the command). Next the numbers of the two channels to connect have to be entered.

When an integer is required, typing only CR means zero.

* Since version 3.0. In earlier versions you may disturb user networks when using this command.

Example : To connect channel 12 with channel 34 on NCU 0 the following sequence has to be executed:

Manual connection:

```
Enter NCU number (-1 terminates) [-1, x]: 0
```

```
Enter channel number [ 0, 95]: 12
```

```
Enter channel number [ 0, 95]: 34
```

where x denotes the maximum NCU number.

Warning : If you connect channels connected to processor links by hardware, you should allocate the corresponding processor. Otherwise a user configuration request could fail (cf. error code -1513).

You should avoid so called 'overswitching' of channels as first switching channel 0 with channel 1 and then also channel 1 with channel 2, for instance. This will result in undefined states.

2. <D> Disconnect channels

Purpose : This command allows to undo existing connections of NCU channels 'manually' for test purposes. The specified channels will be disconnected directly on reset and link level.

Since version 3.0 a warning is printed if user networks have been disturbed by this command.

Usage : Notice the appearing warning and type any key. The rest is analogous to 1.

Warning : The channels to disconnect can belong to a user network. The changes in existing user networks will directly affect the operation and possibly destroy the network. Therefore, this command should not be executed while users are requesting resources from the MultiCluster 2 / SuperCluster or running applications.

3. <S> Show NCU status

Purpose : Gives a tabular overview of all connected channels both of the link and the reset level. Every NCU and every switching level will be displayed on a new screen. The output is sorted in ascending order of physical NCU numbers (cf. Appendix D "Physical Device Numbers").

Usage : Type <S> to step through the pages.

Example : Let us assume that the channels (12,34) and (14,45) are connected respectively. The resulting screen for the NCU 0 link level will look like figure 2.4.

```

NETWORK CONFIGURATION MANAGER   Version: 3.0
                                Copyright (C) 1990, Parsytec GmbH

Physical connections of NCU channels :

NCU 0  link board

12 <> 34  14 <> 45

1 <> 95 : Channel 1 is connected bidirectional to channel 95
1 -> 95 : Output channel 1 is connected to channel 95
1 == xx : Error xx while reading channel 1.
Hit any key

```

Fig 2.4 NCU status example

To determine the physical numbers of the involved processors please consult the Technical Documentation for your system configuration.

Note : The <> sign between two channel numbers indicates a bidirectional connection. Another sign -> indicates an irregular switching which typically occurs by overswitching (cf. warning in topic 1 "<C> Connect channels" above).

4. <P> Check free processors (for SuperClusters only)

Purpose : Check all free processors for the following topics:

- single or multiple bit error in RAM
- detection of bad address access
- deleting of status register flags
- functioning of transputer error setting

If no error is detected on a processor, it becomes deallocated and usable for all users. The green led at the processor is flashed.

If the check fails for a processor, it becomes allocated by the system.

Usage : Type any key to step through the pages.

Warning : This command also checks system allocated processors and may release them afterwards.

Conventions for output :

Each NCU produces one screen in ascending order of its physical number (cf. Appendix D "Physical Device Numbers").

Each entry of the screen represents one single link of a processor.

In clusters with 16 processors (Computing Clusters) you will find 64 entries. The first four entries represent link 0-3 of the first processor in this cluster, the next four entries link 0-3 of the second processor etc.

The possible entries (32 bit hexadecimal number) have the following meanings:

000000xx: If the most significant bit is 0, the processor check succeeded (via this link). xx means the processor node has xx MB RAM available (result of the ident pal).

FFFFFFFF: No processor was found via this link. There can be 3 reasons:

- The processor is allocated by a user.
- The processor was not found.
- The processors link may be faulty.

8000yy00: The most significant bit set means that an error occurred. The **8 bits** of yy have the following meanings if they are set:

Bit 0: The EDC (Error Detection and Correction) Unit of the checked processor detected a single or multiple bit error.

Bit 1: The processor check performs an address access to non existent RAM to test the address error detection of the checked transputer. If this bit is set, the address error was not detected.

Bit 2: After the address error test, the status register reset is tested. If the status register is not reset after reading the status, bit 2 is set.

Bit 3: A program is loaded on the transputer to set the error flag. A dummy access latches the error status to the status register. The lowest bit of the status register should become 0 (transputer error). If not, bit 3 is set.

Bit 4: After the transputer error check is done, the status register should be resetable by reading. If not, bit 4 is set.

5. <R> Reset all NCUs

Purpose : Executing this command will set the C004s on all NCUs in a defined state after confirmation. Thus, all channel connections will be eliminated.

Warning

All existing networks will be destroyed and all users will lose connection to their networks.

Never execute <R> when users are working on the machine !

Note : A warning is produced if a user network is destroyed by this command (since version 3.0).

6. <N> Return a user network

Purpose : If a user forgot to return his network, the processors he configured are not available for other users. The dispatcher can return the network of each user. Choosing "user -1" system allocated resources may be released.

Usage : After typing <N>, you will be asked for the number (1..8) of the user whose network you want to return. Ensure that you have chosen the right user, then enter 'y' for 'yes' or 'n' for 'no'.

Entering 0 for user number has no effect.

Entering -1 will return all resources allocated by the system (internal user number = -1). All system allocated devices and channels are released. The channels connected by the system (function <C>) are **not** disconnected physically !

Warning : Using <N> for user -1 (system) the channel connections will not be removed physically, but the channels are released to be distributed to any user afterwards.

Returning a user network will stop the user application.

7. <U> Display processor usage

Purpose : The usage of all processors is displayed by the physical processor numbers (cf. Appendix D "Physical Device Numbers") in blocks of 32 entries. After each physical processor number, the state entry is printed. A processor may be free (entry "--"), allocated by the system (entry "syst.") or requested by a user (entry "USR i", where i is the user id).

Usage : Type any key to step through the pages.

8. <V> Display usage of administrated devices

Purpose : There is only one type of administrated device, the special processor boards. The usage of these devices is displayed by their physical numbers (cf. Appendix D "Physical Device Numbers") in the same way as the processors (cf. 7. <U>).

Usage : Type any key to step through the pages.

9. <X> Display channel usage

Purpose : The NCU is a multiusable device and its channels may be used by several users. Each NCU is displayed by its physical number (cf. Appendix D "Physical Device Numbers") at an extra screen. All 96 channels are shown by their number followed by "--" (= free), "syst." (= system allocated) or "USR i" (= allocated by user i).

To determine the physical numbers of the NCU channels please consult the Technical Documentation. (There might be some differences between the hardware naming of the NCUs and the physical numbers handled by the NCM !)

Usage : Type any key to step through the NCU pages.

10. <A> Allocate free processors

Purpose : By this command free processors can be allocated. Once allocated free processors by the dispatcher, the NCM will not distribute them as available processors to the users.

Usage : After executing the <A> command, you will be asked to enter the *physical* numbers of the processors (cf. Appendix D "Physical Device Numbers") to allocate (finish with -1). If you choose an allocated processor or an invalid number, "used" resp. "invalid" is typed behind the number. Type any key to remove this information.

When an integer is required, typing only CR means zero.

11. <F> Deallocate processors

Purpose : Allocated processors (with option <A>, <Z> or via processor check) can be released again and will return into the processor pool of the NCM.

Usage : Analogous to <A> option.

12. <Z> Allocate all free processors

Purpose : All free processors become system allocated by this command (cf. command <A>).

13. <Y> Deallocate all processors

Purpose : All system allocated processors (including defective or not existing processors) become free. They can be distributed in a configuration request afterwards (cf. command <F>).

14. Allocate administrated devices

Purpose : It is important to allocate special processor slots if no boards are present. This is done in the same way as described above (option <A>) for application processors. The special processors are called administrated devices here because the NCM treats these processors like the application processors, i.e. they are fully integrated and distributed in the system.

The only difference between special processors and application processors is that the special processors may be individually addressed by a definite number. Thus, it is possible to specify a particular special processor within the application. By way of contrast the application processors have all the same functionality and thus, there is no need to be able to specify a particular physical processor.

Since the logical processor numbers differ from the physical numbers the following table will give an overview of the number mapping to be able to name the processors to be allocated:

0	: Processor 1100
1	: Processor 1200
2	: Processor 1300
3	: Processor 1400
4	: Processor 1500
5	: Processor 1600
6	: Processor 1700
7	: Processor 1800

(cf. section 3.3 "Special Requirements for Processor Numbering" and Appendix D "Physical Device Numbers")

Usage : Analogous to <A> option.

15. <G> Deallocate administrated devices

Purpose : Previously allocated devices (with option) can be released again and will return into the device pool of the NCM. This option is intended for undoing incorrect inputs which have previously been made with the option.

Usage : Analogous to <A> option.

16. <O> Omit random behaviour for partitioning

Purpose : This command is useful only for clustered machines (MC 2/64-4, SC 32, 48, 64, 128, 256, 320), where the NCM works with the partitioning algorithm to map the requested devices to the different clusters.

The partitioning algorithm needs random numbers. The partitioning for dense topologies which have to use a lot of external links may succeed with a special initial value for this topology.

The random behaviour is undesired for hardware testing.

The option <O> may stop random behaviour and you may enter your special initial value.

Usage : After typing <O>, answer the question "Random behaviour desired ?" by entering '0' for 'no' or '1' for 'yes'.

0: You may enter your specific initial value. Partitioning always has the same result now. Different initial values force another partitioning behaviour.

1: The NCM shows the first initial value used. This value may change later on.

17. <Q> Quit

Purpose : To leave the »EXE NCM.control« and to finish the interactive mode of the NCM system interface.

2.2.2 Helios »boot_ncm«

With the Helios tool »boot_ncm« it is possible to boot the Network Configuration Manager, do an initial processor check, allocating resources by the system and disable the partitioning random behaviour.

The »boot_ncm« control functions work in the 'non-interactive' mode so that it becomes usable in shell scripts.

Booting and using control functions may be done by options in a single step.

2.2.2.1 Booting

Booting the NCM first an init program is loaded to switch the main processors' link to the sub processors' link and to detect the type of the main processors (T414 or T800). Afterwards, the corresponding NCM system software is booted.

Location of System Software

Since version 3.0 the system software for Helios is located in a special directory »/helios/ncm«.

In this directory you will also find the executable file »boot_ncm« and a »setdef« shell script.

The naming conventions for the system software files are as follows:

initial program file: ncmi?xxx.bin

NCM for T4 network: ncm4?xxx.bin

NCM for T8 network: ncm8?xxx.bin

where ?xxx names your MultiCluster 2 / SuperCluster system:

? is 'm' for MultiCluster and 's' for SuperCluster.

xxx is the size of your system (number of processors).

For a SuperCluster 64 for example your files are named ncmis64.bin, ncm4s64.bin and ncm8s64.bin.

With these conventions you are able to hold NCM software for several different machines on your »ncm« directory.

Setting up the Default Machine

You will set up your default machine by calling the `»setdef«` shell script by

```
setdef ? xxx      (with the above conventions)
```

Example :

```
cd /helios/ncm
setdef s 64
```

sets up the default machine to SuperCluster 64. The files `ncmis64.bin`, `ncm4s64.bin` and `ncm8s64.bin` will be copied to the default files `ncm_init.bin`, `ncmt4.bin` and `ncmt8.bin` respectively.

Afterwards, you are able to boot the default machine by

```
boot_ncm -b
```

if the files `ncm_init.bin`, `ncmt4.bin` and `ncmt8.bin` will match the default files used by `»boot_ncm«` (displayed via `»boot_ncm -h«`). Ensure that you do not use an older version of `»boot_ncm«` located in `»/helios/bin«`. (The release number of `»boot_ncm«` has to be equal or larger than [901204].)

Otherwise you are able to boot by

```
boot_ncm -i ncm_init.bin -b ncmtx.bin
```

where $x = 4$ for T414 NCUs and $x = 8$ for T800 NCUs.

Boot Procedure

`»boot_ncm -b«` starts up the Network Configuration Manager (NCM) by sending two programs to the Network Configuration Units (NCUs) of a MultiCluster 2 or SuperCluster.

If the NCM is already running, a reboot can only be done by using option `-yes`.

When starting the NCM, a system processor check is performed by default. If this check fails, the number of retries can be got by calling `boot_ncm -h`. The check can be suppressed by using option `-no`.

If the NCM is booted successfully, the following message is printed to standard output:

```
Loading /helios/ncm/ncm_init.bin ...
Successfully booted /helios/ncm/ncm_init.bin.
The NCU processors are of type T8.
```

```
Loading /helios/ncm/ncmt8.bin ...  
Successfully booted /helios/ncm/ncmt8.bin.  
Checking processors ... ok.  
Network Configuration Manager Version 3.0 : Startup and  
Check ok.
```

Boot Options:

```
boot_ncm [-i] [-b] [-l] [-yes] [-no]
```

- i <file_name>** Boot the Network Configuration Manager. You can give a file name for the init program file different from the default one. <file_name> is optional.
- b <file_name>** You can give a file name for the NCM system software different from the default one. <file_name> is optional.
- l <link_no>** You can specify a link for the connection to the NCM network different from the default one.
- yes** The NCM is booted even if the NCM is running already. (All user networks are lost !)
- no** The initial processor check will be skipped.

Example 1 :

```
boot_ncm -b
```

boots the Network Configuration Manager if it is not already running. Default file names are used.

Example 2 :

```
boot_ncm -i ncm_init.bin -b ncmt8.bin -yes
```

boots the Network Configuration Manager on T800 NCUs even if it is already running. The given file names (located in the ncm directory) are used.

WARNING

Reloading and rebooting of the NCM software will undo all existing network configurations and all user programs. Therefore, be sure that no user is running a program on the MultiCluster 2 / SuperCluster when booting and loading the NCM software.

2.2.2.2 Error Situations and Warnings

While loading the system software as described above, the following warnings or error messages may appear after calling »boot_ncm -b«.

1. WARNINGS

If the NCM is already running on the NCU, a second loading would destroy all existing user networks. Thus, the following message is printed on the screen:

```
Network Configuration Manager Version 3.0 is running :  
Use -yes to reboot.
```

If you use option `-yes`, the NCM will be loaded again and all configurations will be reset.

If the NCM `init` program detects a NCU network mixed of T414 and T800, the following warning is displayed:

```
Loading /helios/ncm/ncm_init.bin ...  
Successfully booted /helios/ncm/ncm_init.bin.  
The NCU processors are of mixed type.  
  
Loading /helios/ncm/ncmt4.bin ...  
Successfully booted /helios/ncm/ncmt4.bin.  
Checking processors ... ok.  
Network Configuration Manager Version 3.0 : Startup and  
Check ok.
```

The T800 code never runs on T414. If T414 code is used, it depends on the type of the NCU 0 (administration process NCU) whether the NCM will work or not.

Choose the NCU 0 to be a T414 NCU and use T414 code to run the NCM.

2. ERRORS

If the NCM is not installed completely or no default machine is defined, the following message (or a similar one) may be written to standard error:

```
Loading ncmis16.bin ...  
Successfully booted ncmis16.bin.  
The NCU processors are of type T8.  
  
Loading /helios/ncm/ncmt8.bin ...  
Error reading /helios/ncm/ncmt8.bin :  
Error opening file.
```

If you use »boot_ncm -b« without file names, some default file names have to exist (call »boot_ncm -h« and look for 'Following defaults are used:').

If you already have installed the NCM, simply call `»setdef ? xxx«` (? = 'm' or 's' for MultiCluster 2 or SuperCluster respectively and xxx is the number of processors in your system: 16, 32, 48, 64, 128, ...).

Another way to avoid the problem is to define the file names to use via options `-i` and `-b`.

If the NCM software is not installed for the desired machine (files `ncmi?xxx.bin`, `ncm4?xxx.bin` and `ncm8?xxx.bin` do not exist at `/helios/ncm`), refer to chapter 4 "Installation".

An older version of `»boot_ncm«` could exist on the directory `»/helios/bin«`. Please delete this file or type `»./boot_ncm«` instead of `»boot_ncm«`.

If you get the following error message ...

```
Loading /helios/ncm/ncm_init.bin ...
Error while sending /helios/ncm/ncm_init.bin :
NCU pipeline not complete.
Timeout while sending to link.
```

... there is no way for the host processor to reach the communication subsystem. Thus, the boot code of the NCM can not be sent to the NCU. Several reasons are possible:

- o The specified boot link is wrong. You should change the boot link by using option `-l`.
- o There is no physical connection (no wire or a broken wire) between the host processor link and the ConfigIn link of the MultiCluster 2 / SuperCluster.
- o Other reasons which result in a never responding NCM (no power, no NCU boards inserted, etc.).
- o Same message for `»ncmt8.bin«`: You may have some NCUs with T414 main processor in your system.
- o Perhaps the MultiCluster 2 / SuperCluster is not switched on.
- o An older version of `»boot_ncm«` could exist on the directory `»/helios/bin«`. Please delete this file or type `»./boot_ncm«` instead of `»boot_ncm«`.

If you get the following message ...

```
Loading /helios/ncm/ncm_init.bin ...  
Error switching link level on processor x.
```

... a C004 may have got a protocol error or the power up reset has failed.

Power off the MultiCluster 2 / SuperCluster, wait for some seconds and power on again.

The same message may occur for the reset level.

The following message is displayed if the NCU 0 (administration process NCU) has the wrong type (T414 code loaded on a T800 processor):

```
Loading /helios/ncm/ncm_init.bin ...  
Successfully booted /helios/ncm/ncm_init.bin.  
The NCU processors are of mixed type.
```

```
Loading /helios/ncm/ncmt4.bin ...  
Failed to boot /helios/ncm/ncmt4.bin :  
NCU has wrong processor type. Please use T8 code to boot  
the NCM.  
For mixed systems : Use T4 NCU as NCU 0 and load T4 NCM.
```

The administration process uses floating point operations which are not compatible for T414 and T800. So the NCM software has to be available in the file »ncmt8.bin«.

For mixed systems there may be a way to run the T414 NCM (file »ncmt4.bin«). If you choose the NCU 0 to be equipped with a T414 main processor, the NCM may work with other NCUs having a T800.

2.2.2.3 NCM Control Functions

If a system is not equipped with the maximum number of (special) processors, these 'missing' devices can be allocated. Thus, the NCM does not take these nodes into consideration while distributing the resources.

The random behaviour of the partitioning algorithm can be disabled for testing hardware or to get an optimal partition for a special network topology.

Control Options:

```
boot_ncm [-l] [-ap] [-ad] [-fp] [-fd] [-re] [-rd]
```

-l <link_no> You can specify a link for the connection to the NCM network different from the default one.

-ap <list> The processors given in <list> are allocated by the NCM. They cannot be used later on until they are released. Processors already allocated by a user are not allocated. No feedback is given if a processor was found allocated or a number was invalid.
<list> = <number> [{,|-}<list>]

-ad <list> Other administrated devices (special processors) given in <list> are allocated by the NCM, cf. -ap. No feedback is given if an administrated device was found allocated or a number was invalid. It is important to allocate special processor slots if no boards are present. The special processors are called administrated devices here because the NCM treats these processors like the application processors, i.e. they are fully integrated and distributed in the system.

The only difference between special processors and application processors is that the special processors may be individually addressed by a definite number. Thus, it is possible to specify a particular special processor within the application. By way of contrast the application processors have all the same functionality and thus, there is no need to be able to specify a particular physical processor.

Since the logical processor numbers differ from the physical numbers the following table will give an overview of the number mapping to be able to name the processors to be allocated:

0 : Processor 1100
 1 : Processor 1200
 2 : Processor 1300
 3 : Processor 1400
 4 : Processor 1500
 5 : Processor 1600
 6 : Processor 1700
 7 : Processor 1800

(cf. section 3.3 "Special Requirements for Processor Numbering" and Appendix D "Physical Device Numbers")

- fp <list>** The processors given in <list> are released and returned into the processor pool of the NCM. User allocated processors are not released.
 <list>: cf. -ap
- fd <list>** Administrated devices (special processors) given in <list> are released (cf. -af). User allocated devices are not released.
 <list>: cf. -ap
- re** Random behaviour for partitioning Enable.
- rd <number>** Random behaviour for partitioning Disable.
 <number> is the initial value for computing the random numbers. The behaviour is equal if <number> is the same.

The last two options are useful for clustered machines only (MC 2/64-4, SC 32, 48, 64, 128, 256, 320), where the NCM works with the partitioning algorithm to map the requested devices to the different clusters.

The partitioning algorithm needs random numbers. The partitioning for dense topologies which have to use a lot of external links may succeed with a special initial number for this topology.

The random behaviour also is undesired for hardware testing.

Example 1 :

```
boot_ncm -ap 0,10-12,15
```

allocates the processors 0, 10, 11, 12 and 15 by the system.

Example 2 :

```
boot_ncm -rd 134 -l 1
```

disables the random behaviour of partitioning. The initial value used is 123. The link of the host machine connected to ConfigIn is 1.

2.2.2.4 Exit Codes

»boot_ncm« may terminate with the following **exit codes**:

Succeed:

0 : no error

File:

1 : error opening file
2 : error getting position of readpointer (ftell)
3 : error setting position of readpointer (fseek)
4 : allocation error
5 : error reading file
6 : error closing file

Syntax:

7 : bad command line

Link:

20 : link input error (timeout at input link (ConfigIn))
21 : link output error (timeout at output link (ConfigIn))

40 : link mode change error
41 : link allocation error
42 : link free error
43 : link previous mode error

NCM:

50 : NCM startup failed
51 : error checking processors
52 : error allocating processor
53 : error allocating administrated device
54 : error deallocating processor
55 : error deallocating administrated device
56 : command acknowledge error

2.2.3 ConfigIn Link Protocol

2.2.3.1 Booting

The NCM System Software can be loaded by any workstation which is able to boot a transputer via link.

Three steps have to be done two times (once for the `NCM.init` program and another time for the NCM itself) to boot the NCM:

1. Reset the main transputer of the first NCU connected to the ConfigIn link.
2. Load the code of the program by sending the binary program code via the link.
3. Drive a startup protocol different for `NCM.init` and NCM.

Annotations :

Step 1 : Note the different reset schemes for Parsytec and Inmos transputer boards.

Step 2 : The code is delivered in the directory `helios/ncm` (files: `ncmi?xxx.bin`, `ncm4?xxx.bin` and `ncm8?xxx.bin`)

The binary files are the result of the occam 2 compiler extraction to be loaded without any further protocol. Notice that you cannot use the NCM system software delivered within the `mtool` directory.

Step 3 : First receive the startup result (32 bit integer) from the ConfigIn Link (equal for both programs).

startup code i	result
#12345678	startup ok
$0 \leq i < 1000$	NCU i did not respond
$1000 \leq i < 2000$	timeout switching hardware at link level of NCU i-1000
$2000 \leq i < 3000$	timeout switching hardware at reset level of NCU i-2000
$3000 \leq i < 4000$	communication failure (timeout, crc check failed between NCU main processors i-1001 and i-1000)

4000 <= i < 5000	missing response from NCU main processor i-4000
5000 <= i < 6000	the NCM compiled for T414 has been loaded on a T800 network at least for the administration process of the NCM.

When these three steps are finished, a different protocol has to be handled with the `NCM.init` and the NCM itself (**only if startup succeeded**, i.e. startup result = #12345678).

These protocols are given in an occam-like code:

Further NCM.init protocol (since version 3.0):

```
SEQ
  ConfigIn ! 0
  ConfigIn ? network.processor.type
    -- 0 : mixed NCU network
    -- 1 : T414 NCU network
    -- 2 : T425 NCU network
    -- 3 : T800 NCU network
    -- 4 : T805 NCU network
```

In dependency of the network processor type the file `ncm4?xxx.bin` (type <= 2) or `ncm8?xxx.bin` (type >= 3) has to be loaded.

Further NCM Protocol:

```
ok := FALSE

WHILE NOT ok
  SEQ
    ConfigIn ! processor.check.code
      --> = 1 : perform initial processor check (makes sense for SuperClusters only)
      --> else: skip the initial processor check
    IF
      processor.check.code <> 1
        ok := TRUE
        processor.check.code = 1
        SEQ
          ConfigIn ? number.of.defective.procs
          SEQ defect.proc = 0 FOR number.of.defective.procs
          Config.In ? defective.procs [defect.proc]
          ConfigIn ! code
            --> < 0 : retry check, ok remains FALSE
            --> = 0 : accept defective processors, they are allocated by the NCM,
            --          ok := TRUE
            --> > 0 : number of defective processors to deallocate, ok remains FALSE
        IF
          code > 0
            SEQ
              SEQ count = 0 FOR code
              ConfigIn ! number.of.processor -- to be deallocated
              ok := TRUE
              code = 0
              ok := TRUE
          ConfigIn ? dummy --> = 0
```

Occam conventions:

```
ConfigIn ? integer.variable      : Receiving a (32 bit) integer from the ConfigIn link
ConfigIn ! integer.value        : Sending a (32 bit) integer to the ConfigIn link
a <> b : a not equal to b
```

2.2.3.2 NCM Control Functions

When the NCM is started up, a control function protocol may be driven via the ConfigIn link.

There are two different modes to drive this protocol:

- interactive mode
- non-interactive mode

Changing the mode is possible after the protocol in one mode has been finished.

The interactive mode :

The interactive mode is started up by sending a carriage return (32-bit integer value 13) via the ConfigIn.

The following protocol is a MultiTool server protocol with all control bytes.

A link monitor has to be started at the host machine which collects the output from the ConfigIn link and sends it to the terminal, and to read the keyboard input and send it as integers to the ConfigIn link.

The protocol is finished if the NCM sends a »`tt.terminate`« character (decimal 15 as BYTE).

All menus are generated and all keyboard input is controlled by the NCM.

The communication is interactive.

Non-interactive mode :

The non interactive mode can be used to control the NCM by programs or an operating system.

The protocol consists of three parts:

- Start the non-interactive protocol
- Perform commands
- Finish the protocol

Protocol Start :

```

ConfigIn ! #54494E49
ConfigIn ? x          (integer 32 bit)
ConfigIn ? y          (integer 32 bit)

```

where x.y represents the NCM version number, # means hexadecimal,
! send and ? receive from ConfigIn link.

Perform Command :

```

ConfigIn ! Command.Code
ConfigIn ? acknowledge --> 0: command ok
                   --> -1: invalid command
ConfigIn <-> different command protocols -- see below

```

Protocol Finish :

```

ConfigIn ! 99          --> finish command
ConfigIn ? acknowledge --> = 0

```

Command Protocols (NCM control functions, non-interactive mode) :

```

Command.Code = 11    manual channel connect

connects channels »channel from« and »channel.to« at NCU
»NCU.number«. MAX.NCU is the number of NCUs in the system.

protocol after Command.Code & acknowledge:

    ! INT NCU.number
    -- -1: terminates loop
    -- 0 <= NCU.number < MAX.NCU: NCU.number
    -- else: like -1
    WHILE -1 < NCU.number < MAX.NCU
    SEQ
        ! INT channel.from
        ! INT channel.to
        ! INT NCU.number

Command.Code = 12    manual channel disconnect

same protocol and corresponding function as manual connect

Command.Code = 14    reset all NCUs

Command.Code = 17    allocate processors

MAX.PROCESSOR is the number of application processors
available in the system.

    ! INT proc.nr
    -- 0 <= proc.nr < MAX.PROCESSOR: valid
    -- < -1 or >= MAX.PROCESSOR: invalid (skip)
    -- = -1: terminate loop

```

```

                                WHILE proc.nr <> -1
                                  ! INT proc.nr

Command.Code = 18    deallocate processors
                    same protocol as allocate processors

Command.Code = 20    allocate all processors
                    ? proc.nr
                    WHILE proc.nr <> -1
                    ? proc.nr
                    -- >= 0: allocated processor
                    -- -1: terminate loop

Command.Code = 21    release all processors
                    same protocol as allocate all processors

Command.Code = 22    allocate administrated devices (special processors)
                    same protocol as allocate processors with
                    MAX.ADM instead of MAX.PROCESSOR.
                    MAX.ADM is the number of administrated devices
                    available in the system (cf. Appendix D "Physical
                    Device Numbers").

Command.Code = 23    deallocate administrated devices
                    same protocol as allocate administrated devices

Command.Code = 30    disable / enable random behaviour of partitioning algorithm
                    ! func
                    -- 0: disable
                    -- <> 0: enable
                    IF
                    func = 0
                    SEQ
                    ! starting.value
                    ? used.starting.value

Command.Code = 35    return a user network
                    ! user.id
                    -- 1..8: return network of user
                    -- -1: deallocate all system allocated
                    resources (without disconnect)

```

For exact descriptions of the above commands compare the interactive command description of the MultiTool »EXE NCM.control« in chapter 2.2.1.3 "The NCM Monitor".

The User Interface Software

The user interface software for the MultiCluster 2 / SuperCluster (MC 2 / SC) has the task to receive user requests, translate them into a defined protocol and send them to the Network Configuration Manager (NCM) which represents the system software running on the MC 2 / SC.

Another task is to receive messages and data from the NCM and display them to the user. These messages may be error or warning messages, for instance.

The main task, however, is to help the user to configure a network on the MC 2 / SC and finally to load a program onto this network. The necessary actions to proceed are done automatically and include a comprehensive communication with the NCM.

To be able to perform such tasks, the interface software has to run on a host system which has a link connection to the MC 2 / SC. This host system may be a SUN, a PC with a plugged in module, or something equivalent with at least one transputer.

The NCM in its turn administrates the processor resources of the MC 2 / SC systems and reacts upon the messages sent by the interface software. A detailed description of the operation of the NCM was given in chapter 2.

Note that the NCM does not know which kind of interface software is talking, due to the fact, that the communications protocol is standardized. There is of course no difference for the NCM between a network request for a MultiTool occam PROGRAM and a network request for Helios.

Thus, the interface software has to extract only the structural information of the desired network and to send this data to the NCM.

As mentioned in chapter 1 the MultiCluster 2/32-2 NCU and the MultiCluster 2/64-4 NCU need a special treatment to meet the hardware requirements of these machines. In the particular following sections the user will be informed which interface software has to be used for which systems concerning network configuration requests.

The two kinds of interface software allowing configuration requests for a *MultiTool user* are:

The »EXE Config and Boot Network« and the extended occam 2 compiler utilities. The first is used with the MultiCluster 2/16-1 NCU and the SuperCluster, the latter is used with the MultiCluster 2/32-2 NCU and the MultiCluster 2/64-4 NCU.

A *Helios user* may request a network by running the program »hconfig«. Afterwards, Helios can be booted on the configured network.

3.1 USER INTERFACE SOFTWARE FOR MULTITOOOL

Basically, there are two kinds of software running within the MultiTool environment, which may serve as an interface to the MultiCluster 2 / SuperCluster.

One is the »EXE Config and Boot Network«. This tool will help the user to request for several services of the NCM. The different features of this program are explained in section 3.1.2.

The other is a modified version of the occam 2 compiler utility, which is extended by an additional function called [CONFIG] replacing the first function (<cr> new configuration and boot network) of the »EXE Config and Boot Network«. This special compiler is necessary for the MC 2/32-2 and MC 2/64-4 only and will be described in section 3.1.3.

Which software you have to use for which purpose is explained below for the different system configurations.

At the end of this section some example programs will illustrate how to run PROGRAMs on the MultiCluster 2 / SuperCluster.

3.1.1 Machine Dependent Network Configuration

Note that the machine dependencies only refer to the process of *configuring a network*, which is the most important operation of the NCM. The interface software extracts the network information from the occam PROGRAM to configure and sends it to the NCM.

Depending on the kind of machine the NCM proceeds in two different ways. On the MultiCluster 2/16-1 NCU and SuperCluster the NCM checks the number of available resources and then tries to configure the network topology. A message of the configuration attempt is returned to the user interface software.

The NCM of the MultiCluster 2/32-2 NCU and MultiCluster 2/64-4 NCU also checks the number of free resources, but then continues with computing a link mapping for the received network request according to the hardware requirements. This modified link configuration does not change the structure of the network but the physical link numbers.

The computed link mapping of course will influence the program code, therefore, the compiler has to be called to recompile the program with the modified link numbers. To simplify and automate the necessary processes an extended occam 2 compiler is delivered with those machines.

3.1.1.1 MultiCluster 2/16-1 NCU and SuperCluster

All users connected to a MultiCluster 2/16-1 NCU or a SuperCluster have to run the »EXE Config and Boot Network« to configure a network and to load an occam PROGRAM automatically. This is done by the first option of the menu (<cr> New Configuration and Boot Network). The exact usage of this tool will be presented in section 3.1.2.

The extended occam 2 compiler utilities (described in section 3.1.3) are not included in the delivery version of the user interface software for this group of machines because there is no need to use the reconfiguration features of this compiler version.

The common occam 2 compiler utilities may be used instead in the known way. There are no limitations or changes in using this compiler.

Summary:

- Write your occam PROGRAM according to the rules of section 3.3.
- Compile the PROGRAM as usual.
- Configure the network and load the program in one step using the »EXE Config and Boot Network«.
- Reload your program as often as you like with the occam 2 compiler utilities function [LOAD NETWORK].

3.1.1.2 MultiCluster 2/32-2 NCU and /64-4 NCU

All users connected to a MultiCluster 2/32-2 NCU or a MultiCluster 2/64-4 NCU have to use the extended occam 2 compiler utilities to configure a network and to load an occam PROGRAM.

The only difference to the common occam 2 compiler utilities is the additional function [CONFIG]. It will allow you to send a configuration request for your program to the NCM.

The section 3.1.3 will give you all details about this version of the compiler.

Note that you can not use the first function of the »EXE Config and Boot Network« to configure and boot your network.

3.1.2 The »EXE Config and Boot«

According to the name of the »EXE Config and Boot network« the main task of this interface program is to configure a network and to boot/load an occam program. However, this first function should be used with MultiCluster 2/16-1 NCU and SuperCluster machines only. Users of the other MultiCluster machines have to use the extended occam 2 compiler utilities instead to configure and boot.

The remaining features of this tool, however, may be used with any machine.

The first thing to do is to connect your host transputer with the MultiCluster 2 / SuperCluster. The location of the particular connectors and plugs is described in the technical documentation.

At the MC 2 / SC you choose one of the user entries and at your host system you take the link which will serve as the network entry. This link connection is used both for communication with the NCM and as the entry into your network.

Now you can [GET CODE] the »EXE Config and Boot network«, place the cursor on the foldset of a compiled occam PROGRAM and [RUN] the EXE. The menu shown in figure 3.1 will appear on the screen.

```

+-----+
| Network Configuration Manager |
| MultiTool User Interface     |
| Version 3.0                   |
| Copyright (C) 1990, Parsytec GmbH |
+-----+

<cr>:  Configure and Boot Network
<s>:   System Status
<d>:   Display Configured User Graph
<r>:   Return Network to NCM
<l>:   Link into Machine (actual link is 2)
<q>:   Quit

Please choose an option:
```

Fig3.1 »EXE Config and Boot network« startup menu.

Before you continue you should check whether the displayed link number (link into MultiCluster 2 / SuperCluster) is identical with the link number of the network entry of your host transputer. If not, you have to use the <l> option to change the link number.

At any time in the last line of the screen the EXE informs about the possible actions. All actions are key sensitive. That means the EXE waits only for a single key and does not wait for an extra carriage return after any action.

Other features of this tool set:

- This EXE is also used to *return a network* to the system. After returning a network to the system, the transputers are free for use in a network of any user who requests a new network.
- The EXE returns a *system status* of the MultiCluster 2 / SuperCluster. The status of the MultiCluster 2 / SuperCluster includes the distribution of the processors among the users, the number of free processors and data about the current or last established network.

1. <cr> Configure and Boot Network

Note : This function should not be used with the MultiCluster 2/32-2 NCU and MultiCluster 2/64-4 NCU since there is no way to reconfigure and to recompile the network/PROGRAM. Users of these machines should use the extended occam 2 compiler utilities instead (see section 3.1.3).

Purpose : This function will take the selected, compiled PROGRAM foldset and search for the descriptor fold previously generated by the compiler. This fold is analyzed and the resulting network structure for the PROGRAM is sent to the NCM via the specified link.

If enough processors are free, the Network Configuration Manager computes an internal representation of the network and programs all link switches for this new network (without changing the network of other users).

If the configuration succeeds, a usual transputer network is available and the code is booted onto this network via the same link. If the first booting fails, the EXE repeats the booting up to 3 times. The resulting screen is shown in figure 3.2.

If the user configured a network before, an implicit return of this old network is done.

```
Network Configuration Manager
MultiTool User Interface
Version 3.0
Copyright (C) 1990, Parsytec GmbH

SuperCluster : User Id. : 1 Channels into network : 64, 65
MultiCluster 2 : User Id. : 1 Channels into network : 64

Partitioning succeeded after 0.1 seconds.

Network configured ok.
Network loaded ok.

hit any key ...
```

Fig 3.2 Successful configuration of a network.

The numbers 1 and 64 in figure 3.2 are examples and may vary according to the user entry.

Error and failure events are described in the Error section below.

Usage : In order to configure a network for your program you have to place the cursor on the compiled occam PROGRAM fold before you start the »EXE Config and Boot Network«.

When the menu is displayed, you have to set up the load link (first user entry) and then to press <cr> (see figure 3.1 <1>). This will allow the EXE to search for the descriptor fold within the PROGRAM foldset and analyze it. The network information is extracted and the result sent to the NCM.

Additional information needed by the NCM about the connection to the host and some other parts which are not contained in the descriptor fold is supplied by ordering the PLACED PAR and special processor numbering (cf. section 3.3 "Special Requirements for Processor Numbering").

After proceeding the steps above successfully, your program will run at the MultiCluster 2 / SuperCluster and a conventional network with a link entry is available.

Note : If you want to boot the same structure again, it is not necessary to use »EXE Config and Boot Network«.

As long as the structure is not returned to the system you can reboot your program with the [LOAD NETWORK] utility function of the occam compiler.

NOTE

If you do not need your network any longer, return your resources to the system. Hence someone else can use these resources.

Errors and Warnings : The first action of the »EXE Config and Boot network« is *ringing* (see section 2.1.1.1) the Network Configuration Manager.

When ringing the NCM, the EXE sends a reset signal on the link to the MultiCluster 2 / SuperCluster.

The subprocessor on the NCU board detects this reset signal (which is longer than the reset in the booting procedure) and informs the main processor via event signals. Then the NCM connects itself to the user.

There are several failures which can prevent the NCM to recognize the reset signal:

- the user enters a wrong link number after starting the EXE
- the input link is not connected correctly
- the system software is not loaded

Figure 3.3 illustrates how the screen looks like. After removing the source of error, you can retry to ring <r> or abort the EXE <a>.

```

+-----+
| Network Configuration Manager |
| MultiTool User Interface    |
| Version 3.0                  |
| Copyright (C) 1990, Parsytec GmbH |
+-----+

Establishing a Connection to the Network Configuration Manager
via link 1 for a system status request:
.....

The Network Configuration Manager does not respond!

(A)abort or (R)etry ?

```

Fig 3.3 Network Configuration Manager does not respond.

Some failure messages of the NCM depend on the desired network to configure.

In a first step the NCM allocates the processors for the network. If not enough processors are free, you will get the following message on the screen:

```

+-----+
| Network Configuration Manager |
| MultiTool User Interface    |
| Version 3.0                  |
| Copyright (C) 1990, Parsytec GmbH |
+-----+

SuperCluster : User Id. : 1 Channels into network : 64, 65
MultiCluster 2 : User Id. : 1 Channels into network : 64

Not enough processors for new configuration.

 16 processors are free.

Your network needs 32 processors.

hit any key ...

```

Fig 3.4 Not enough processors are free for the requested network.

Another failure may occur on clustered machines if the NCM is not able to find a correct partition for the network. This may happen if the structure cannot be mapped into the clusters because the actual distribution of free processors is inappropriate for this network. The screen looks similar to figure 3.5; the error code is -1501. You will see the message 'partitioning failed' on the screen.

Another failure (error code -103) occurs if no free link exists for the first processor in the network to get a connection to the host (cf. section 3.5 "User Entries" and check your program).

If the booting does not succeed even after 3 trials, you will get a message including the number of the processor which cannot be booted (if you extracted the network in DIAGNOSTIC mode). Check the program fold for all requirements given in sections 3.3-3.5 and also check the connections of the input link and the special processors.

```

+-----+
| Network Configuration Manager |
| MultiTool User Interface    |
| Version 3.0                  |
| Copyright (C) 1990, Parsytec GmbH |
+-----+

SuperCluster : User Id. : 1 Channels into network : 64, 65
MultiCluster 2 : User Id. : 1 Channels into network : 64

Configuration failed.

Error Code : -804 : Device(s) (e.g. special processors) not available.

Processors returned to the system pool.

:
:
hit any key ...
:
```

Fig 3.5 Request for a special processor which has already been accessed by another user.

A number of errors that do not occur very often are given by error codes of the NCM. To interpret these codes use Appendix C.

2. <s> System Status

Purpose : With this function you can get information about the following topics:

1. Your own User Id.
2. Your channel (NCU channel) into the network.
3. Number of all free processors in the system (and the distribution into the clusters for clustered machines).
4. The available and not used special processor boards.
5. The processor usage of all users.
6. Processor Mapping (user processor number to physical processor number within the MultiCluster 2 / SuperCluster) for your own network.
7. Connect information of the NCU channels switched for your network.

Usage : Choosing this option will produce a warning on the screen. The user has to realize that a status request will disconnect his network for the time of request*. The user may return to the main menu or continue by pressing <s>. This yields the following figure on the screen:

```

+-----+
| Network Configuration Manager |
| MultiTool User Interface     |
| Version 3.0                  |
| Copyright (C) 1990, Parsytec GmbH |
+-----+

SuperCluster : User Id. : 1 Channels into network : 64, 65
MultiCluster 2 : User Id. : 1 Channels into network : 64

Number of free application processors : 2

Free special processor boards : 1100, 1200, 1300, 1400.

User Id.:      1  2  3  4  5  6  7  8
Used processors: 6  7  0  3  7  4  3  0

:
:
:
Press <s> for processor mapping and NCU connection tables,
any other key returns ...

```

Fig 3.6 NCM Status report.

* The link into your network (your single link into the MultiCluster 2) has to be used for a status request. Ringing the NCM processor will disturb your program running on the configured network. At the SuperCluster the second

In figure 3.6 you see the information of the topics 1-5.

For topic 5 the users are listed by their User Id. (which is determined by the connection host - MultiCluster 2 / SuperCluster) and below the number of assigned processors of every user are displayed.

Pressing the key <s> for a third time yields the table of processor mapping (user (logical) and physical processor numbering):

```
Processor Mapping
Your processors are mapped to the following system processors
1300 -> 34      1 -> 24      2 -> 25      3 -> 26      4 -> 27
5 -> 28      6 -> 29

User processor number -> Physical processor number

:
:
press any key to continue
```

Fig 3.7 Mapping of logical processor numbers to physical.

'User processor number' is the notation of the user in the PLACED PAR or in the resource map.

The 'Physical processor number' is the physical number of the processor listed in Appendix D "Physical Device Numbers".

The physical number of special processors is given by a number greater or equal than the number of MultiCluster 2 / SuperCluster processors.

user entry may be used for a status request without disturbing the network (if the entry is not used for the network).

Typing another key will show the channel connections of each NCU (figure 3.8) beginning with the Top NCU (first NCU in main processor pipe).

To interpret these connections you need detailed information about the hardware of the MultiCluster 2 / SuperCluster (cf. Technical Documentation)*.

```
NCU connection tables :
NCU 0
 13 <> 28    29 <> 30    44 <> 12    60 <> 14    76 <> 64
NCU 1
 12 <> 76    28 <> 77
:
hit any key ...
```

Fig 3.8 Connection table for NCU channels.

Warning : If you have already established a network and ask for the system status, **the NCM disconnects the network**. The same network is available for you after the status report but you have to reload the program which can be done with the [LOAD] function of the occam 2 compiler utilities. The network will be cancelled when you give it back to the system with the option <r>.

Errors : cf. 1.<cr> Configure and Boot Network, "Ringing Error" shown in figure 3.3 and described in the text before the figure.

* 13 <> 28 for example means that the NCU channel 13 is connected to the NCU channel 28.

3. <d> Data of Configured Graph

Purpose : As the previous <s> option this is a status information, too. The different screen pages which will be produced give a compressed view on the graph configured by the NCM, based on the user's program.

```

Graph realised by the NCM : u+ d- c- l- nodes: 0- 15
node link node link node link node link node link node link node link
* 0 0 : 1300 0 1300 0 : * 0 0 1 0 : 1300 1 2 0 : 1300 2
  1 : 1 0 1 : 3 0 1 : 5 0
  2 : 2 0 2 : 4 0 2 : 6 0
  3 : 3 : 3 :

3 0 : 1 1 4 0 : 1 2 5 0 : 2 1 6 0 : 2 2
  1 : 1 : 1 :
  2 : 2 : 2 :
  3 : 3 : 3 :

<u> <d> <c> <l> <ret> <blank> <p> <b> <n> <h>elp <q>uit :

```

Fig 3.9 Graph Display with default representation (earlier versions may display figure 3.16 as default).

Usage : After pressing the <d> key two times, a table will appear like shown in figure 3.9. Each block in this table gives information about a node and its four links. For each of these links the connected node and link is displayed.

In the default representation a block is structured as follows:

```

node link node link
nnnn 0 : nnnn 1
      1 : nnnn 1
      2 : nnnn 1
      3 : nnnn 1

```

Fig 3.10 Default block layout u+ d- c- l-

where nnnn stands for a node named by the user and 1 for a link number (0..3). If an asterisk appears in nnnn, the node was defined by the NCM and has no name given by the user.

In order to maximize the clearness of the compact screen layout some information may be suppressed. This is controlled by the settings of 4 different flags (u,d,c,l). The state of these flags is shown in the first line of the screen.

A plus sign after the flag name indicates that the flag is set (true) and a minus sign that it is reset (false). In the following explanations an x sign after the flag name indicates don't care.

The default layout (figure 3.9), for instance, has most informations disabled. Thus, the status line contains u+ d- c- l-.

- The **u flag** controls whether nnnn in the block above (figure 3.10) denotes the internal NCM node number (u-) or the logical node number (u+), i.e. the user name (number) used in an occam PROGRAM configuration section.

If there is no corresponding logical node number for an NCM node number, an asterisk in front of the node number will be printed. For example the user entry which is treated as NCM node 0 is never specified in a program.

- The **d flag** controls whether the mapping of the nodes onto the physical devices is displayed or not. With d+ a block looks like that:

```

node link node link
nnnn 0 : nnnn 1
      1 : nnnn 1
XXXX 2 : nnnn 1
mmmm 3 : nnnn 1

```

Fig 3.11 Block layout with ux d+ c- l-

where nnnn denotes a node number and mmmm the number of a physical device (cf. Appendix D "Physical Device Numbers"), 1 a link number and XXXX a string which specifies the kind of device. XXXX may have the following values:

```

USER   for a user entry (mmmm is the user id)
PROC   for an application processor
SPEC   for a special processor

```

If a single - sign occurs instead of XXXX and mmmm, the NCM detected an error before mapping the nodes onto physical devices.

The block below for example has to be interpreted as follows: node 3 (user denotation) is mapped to the physical processor 26 and link 0 of node 3 is connected to link 0 of node 1 (also user denotation).

```

node link node link
   3  0 :    1  0
       1 :        1
PROC  2 :        1
   26 3 :        1

```

Fig 3.12 Example for block layout with u+ d+ c- l-

- The **c** flag controls whether a cluster information for the physical devices (the node is mapped to) is given. The example in the figure below shows that PROC 26 is located in cluster 0.

```

node link node link
   3  0 :    1  0
c1 0  1 :        1
PROC  2 :        1
   26 3 :        1

```

Fig 3.13 Example for block layout with u+ d+ c+ l-

If an entry **c1 -** occurs, the NCM detected an error before partitioning.

Note that the cluster information is only interesting for clustered machines like the MultiCluster 2/64-4 NCUs or SuperClusters with more than 16 processors. For other machines the cluster number is always 0.

- The **l** flag controls whether the mapping of logical links to physical links is displayed (only interesting for MultiCluster 2/32 and /64).

The example in figure 3.14 helps to illustrate the link mapping. First of all note that all link numbers on the screen are physical link numbers (in the l+ mode).

```

node link node link
   1  1 : 1300  1
c1 0  2 :    3  0
PROC  0 :    4  0
   24 3 :

```

Fig 3.14 Example for block layout with u+ d+ c+ l+

The first link column of a block is composed of 4 positions for the logical link numbers. The uppermost field is reserved for information about the logical link 0 (a link 0 in the user denotation of the program) and the lowermost is reserved for information about the logical link 3.

The entry in one of the 4 positions is the physical link number to which the logical link belonging to this position is mapped. Thus, in our example the logical link 0 is mapped to the physical link 1. Link 1 is mapped to link 2 and link 2 to link 0. The logical link 3 is not used and is mapped to the remaining physical link 3.

The second link column also contains physical links. So the physical wiring is printed if `1+` flag is set.

If a `-` sign occurs instead of a physical link number, the NCM detected an error before mapping the logical links to physical links.

Note that the information of the link mapping is only interesting for machines which require link renumbering like the MultiCluster 2/32-2 NCUs and /64-4 NCUs. For the other systems the mapping is one-to-one.

To change the value of a flag simply press the key corresponding to the name of the flag (enter `<t>` for true (to set the flag) or `<f>` for false (to reset it) for versions earlier than 2.3) which will toggle the state of the flag.

Only 16 information blocks fit on the screen at the same time. To process graphs with more than 16 nodes the user may browse through the graph with the keys `<ret>`, `<blank>` (which denotes the space bar), `<p>`, `` and `<n>`. On some hosts you may also use the cursor keys. The range of nodes displayed on the current screen is shown in the status line. The default screen will show the nodes 0 through 15.

A **help screen** for the available functions and their keystrokes is displayed after pressing the `<h>` key.

Example : The following example will help to understand the relation between the user program, the resulting graph and the displayed information from the NCM.

First let's have a look at a typical occam 2 program. Only the configuration section of the program is shown. It will form a binary tree structure with 7 nodes:

```

{{{ PROGRAM T8 BinaryTreeStructure
  {{{F T4 BinaryTreeStructure

  VAL TreeHeight IS 2 :

  ... SCs

  {{{ ChannelDef's
  VAL LinkIn IS [ 4, 5, 6, 7 ] :
  VAL LinkOut IS [ 0, 1, 2, 3 ] :

  [2 << TreeHeight] CHAN OF ANY EdgeDown :
  }}}

  {{{ Placements
  PLACED PAR

  PROCESSOR 1300 T8
  PLACE EdgeDown[1] AT LinkOut [1] :
  PLACE EdgeDown[2] AT LinkOut [2] :
  StartProcess (EdgeDown[1], EdgeDown[2])

  PLACED PAR j = 1 FOR TreeHeight-1
  PLACED PAR i = ((1<<j)-1) FOR (1<<j)
  PROCESSOR i T8
  PLACE EdgeDown[i] AT LinkIn [0] :
  PLACE EdgeDown[(i+i)+1] AT LinkOut [1] :
  PLACE EdgeDown[(i+i)+2] AT LinkOut [2] :
  InnerProcess ( EdgeDown[i], EdgeDown[(i+i)+1], EdgeDown[(i+i)+2] )

  PLACED PAR i = ((1<<TreeHeight)-1) FOR (1<<TreeHeight)
  PROCESSOR i T8
  PLACE EdgeDown[i] AT LinkIn [0] :
  FinalProcess ( EdgeDown[i] )
  }}}
  }}}
  }}}

```

As a special processor the GDS is used with the logical node number 1300. The other nodes get the numbers from 1 through 6

The corresponding chart of the network is shown in figure 3.15. This picture contains all available information on a network.

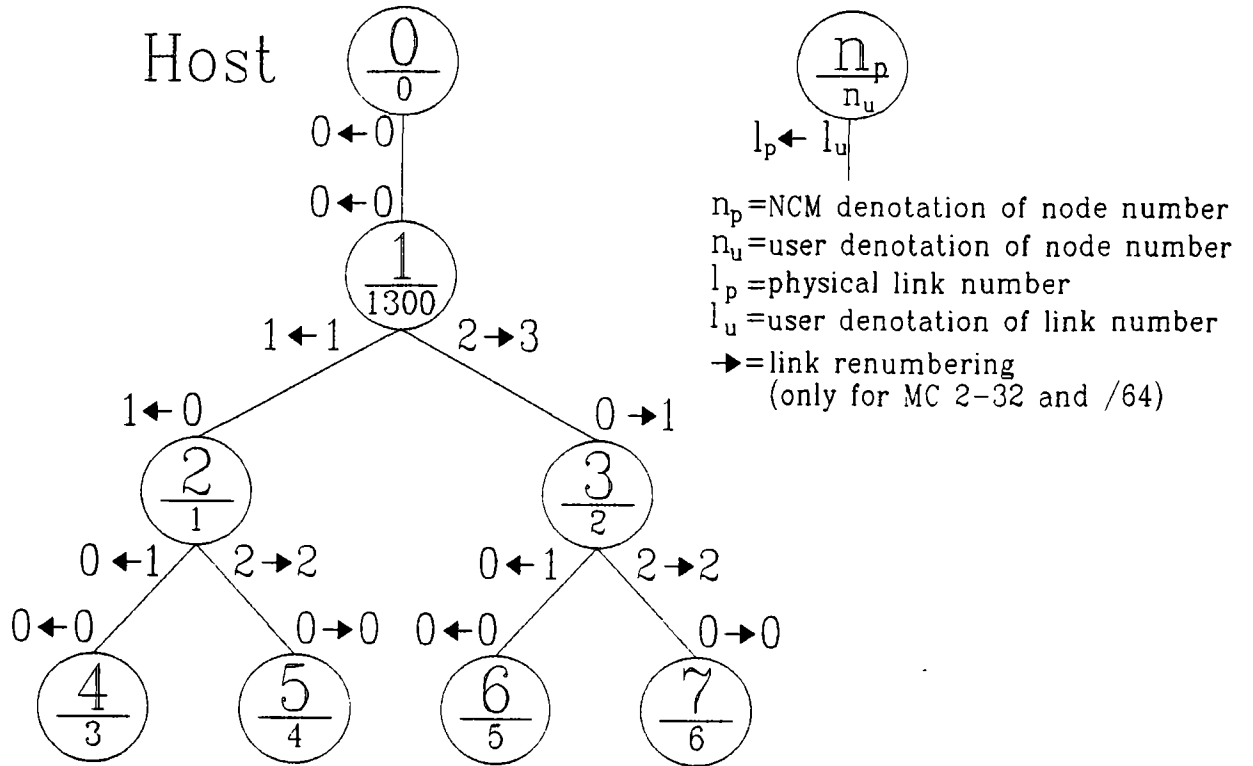


Fig 3.15 Chart of the user network.

Finally the following pictures show the corresponding screens with different flag settings. The default picture which is displayed immediately after choosing option <d> is shown in figure 3.9.

```

Graph realised by the NCM : u- d- c- l- nodes: 0- 15
node link node link node link node link node link node link node link node link
 0 0 : 1 0 1 0 : 0 0 2 0 : 1 1 3 0 : 1 2
 1 : 1 : 2 0 1 : 4 0 1 : 6 0
 2 : 2 : 3 0 2 : 5 0 2 : 7 0
 3 : 3 : 3 : 3 :

 4 0 : 2 1 5 0 : 2 2 6 0 : 3 1 7 0 : 3 2
 1 : 1 : 1 : 1 :
 2 : 2 : 2 : 2 :
 3 : 3 : 3 : 3 :

<u> <d> <c> <l> <ret> <blank> <p> <b> <n> <h>elp <q>uit :
    
```

Fig 3.16 Graph display with the NCM node denotation (node index).

```

Graph realised by the NCM: u+ d- c- l+ nodes: 0- 15
node link node link node link node link node link node link node link node link
* 0 0 : 1300 0 1300 0 : * 0 0 1 1 : 1300 1 2 1 : 1300 3
  1 : 1 1 0 : 3 0 0 : 5 0
  2 : 3 2 1 2 : 4 0 2 : 6 0
  3 : 2 : 3 :

3 0 : 1 0 4 0 : 1 2 5 0 : 2 0 6 0 : 2 2
  1 : 1 : 1 :
  2 : 2 : 2 :
  3 : 3 : 3 :

<u> <d> <c> <l> <ret> <blank> <p> <b> <n> <h>elp <q>uit :
    
```

Fig 3.17 Graph display with logical node numbers and additionally link mapping of logical links to physical links.

```

Graph realised by the NCM: u+ d+ c+ l+ nodes: 0- 15
node link node link node link node link node link node link node link node link node link
* 0 0 : 1300 0 1300 0 : * 0 0 1 1 : 1300 1 2 1 : 1300 3
cl 0 1 : cl 0 1 : 1 1 cl 0 0 : 3 0 cl 0 0 : 5 0
USER 2 : SPEC 3 : 2 1 PROC 2 : 4 0 PROC 2 : 6 0
  1 3 : 2 2 : 24 3 : 25 3 :

3 0 : 1 0 4 0 : 1 2 5 0 : 2 0 6 0 : 2 2
cl 0 1 : cl 0 1 : cl 0 1 : cl 0 1 :
PROC 2 : PROC 2 : PROC 2 : PROC 2 :
  26 3 : 27 3 : 28 3 : 29 3 :

<u> <d> <c> <l> <ret> <blank> <p> <b> <n> <h>elp <q>uit :
    
```

Fig 3.18 Graph display with additionally node to cluster assignment.

Warning : If you have already established a network and ask for the data of the configured graph, **the NCM disconnects the network**. The same network is available for you after the data report but you have to reload the program, which can be done with the [LOAD] function of the occam 2 compiler utilities. The network will be cancelled when you give it back to the system with the option <r>.

Errors : cf. 1.<cr> Configure and Boot Network, "Ringing Error" shown in figure 3.3 and described in the text before the figure.

4. <r> Return Network to System

Purpose : Tells the NCM that the processors of the network are no longer needed. The processors return to the processor pool and can be accessed by other users.

Usage : Every user keeps his network until he explicitly returns it to the system or requests for a new network. If the user starts the request for a new network, the old one is returned automatically.

If you use the »EXE Config and Boot network« only to return your network, it is not necessary to place the cursor on the program fold when you start the EXE.

Choosing the correct link (with <l>), however, is very important. The resulting screen is shown in the following figure:

```

+-----+
| Network Configuration Manager |
| MultiTool User Interface     |
| Version 3.0                   |
| Copyright (C) 1990, Parsytec GmbH |
+-----+

SuperCluster : User Id. : 1 Channels into network : 64, 65
MultiCluster 2 : User Id. : 1 Channels into network : 64

Processors returned to system pool.

:
:
hit any key ...
:
```

Fig 3.19 Returning the processors used to the NCM resource pool.

Warning : If you don't return your network to the system, the processors will be marked as used for the NCM even if you shut down your host system.

Therefore, **don't forget to return the network** to place the resources you have used at another user's disposal.

Errors : cf. 1.<cr> Configure and Boot Network, "Ringing Error" shown in figure 3.3 and described in the text before the figure.

5. <1> Link into MultiCluster

Purpose : With this option you have to specify via which link the communication with the MultiCluster 2 / SuperCluster takes place. This link also is the connection to the configured network afterwards.

Note : All functions of this tool refer to this link number. If you enter a wrong link number, appropriate error messages will be printed while using the other functions.

The proposed connections for the workstations in the Technical Documentation use link 2 which is the default link number after starting the EXE.

The default link number is shown on the screen at the = sign (cf. figure 3.20).

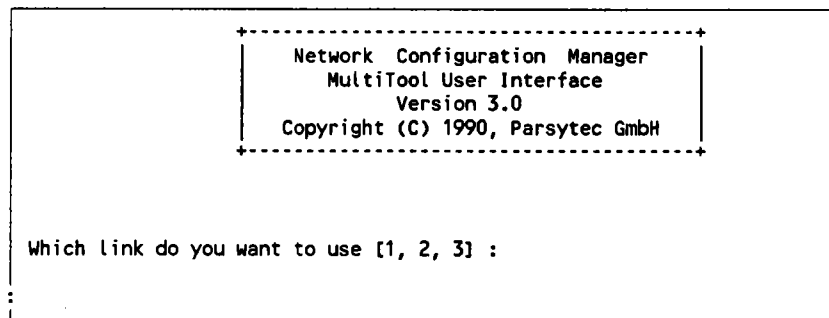


Fig 3.20 Network boot link set-up menu.

6. <q> Quit

Purpose : To leave the »EXE Config and Boot network«.

3.1.3 The Extended Occam 2 Compiler Utilities

If you have a look at the code information fold of the extended occam 2 compiler utilities (figure 3.21), you will detect the new function [CONFIG].

```
Press [EXIT FOLD] to resume editing

((( Code information fold
1 [CONFIG]          - config PROGRAM for transputer network
2 [COMPILE]        - compile current and nested foldsets
3 [EXTRACT]        - extract code and put into fold set
4 [LOAD NETWORK]   - export code to transputer network
5 [RE-COMPILE]     - Use descriptor fold for parameters
6 [COMPILATION INFO] - display compilation information
7 [MAKE FOLDSET]   - make compilation fold
8 [SEARCH/REPLACE] - search/replace text string
9 [ ]              - not used
0 [MAKE COMMENT]   - make comment fold around current fold
)))
```

Fig 3.21 Code information fold for the extended occam 2 compiler utilities.

This function is the automatic equivalent to the manual configuration in conventional transputer systems. It will take the necessary steps to switch the links according to the specifications in the user PROGRAM.

Note that this extended function is delivered only for the MultiCluster 2/32-2 NCU and MultiCluster 2/64-4 NCU. The reason for a modified compiler for these machines is a special hardware configuration.

These machines are not able to connect an 'even' with an 'odd' link. Therefore, the link numbers in occam PROGRAMs are treated as logical links and mapped to the physical links. In this way a (link0, link1) connection in an occam program may be mapped to a physical (link0, link0) connection which is functionally equivalent (cf. chapter 1.3.2 "MultiCluster 2/32-2 NCU and MultiCluster 2/64-4 NCU). This link mapping is done by the Network Configuration Manager (NCM) which is the only process which 'knows' the hardware configuration of the particular MultiCluster 2.

In order to work with the MultiCluster 2 a link connection between your host transputer and the MultiCluster 2 has to be established. At the host transputer you should select the link which will serve as the link entry into the network. At the MultiCluster 2 you choose one of the user entries (L0-L7 at the rear panel). This link connection will be used both for data transfer with your network and for communication between the compiler and the NCM.

The link number (config link) which you select for your host transputer has to be made available to the compiler function [CONFIG]. This is done by entering the Toolkit fold. There you will find the fold "Transputer config parameters" as shown below.

```
{{{  Transputer config parameters
VAL  link           IS 2 : -- 1, 2, 3
VAL  compile        IS TRUE :
}}}
```

Change the link number according to the selection you have made. The "compile" parameter is explained later on. If you do not find the above named folds or their contents respectively, you should add it to the toolkit fold.

Note that the chosen link should be the same as in the "Transputer loader parameter" fold to load your program to the configured network afterwards.

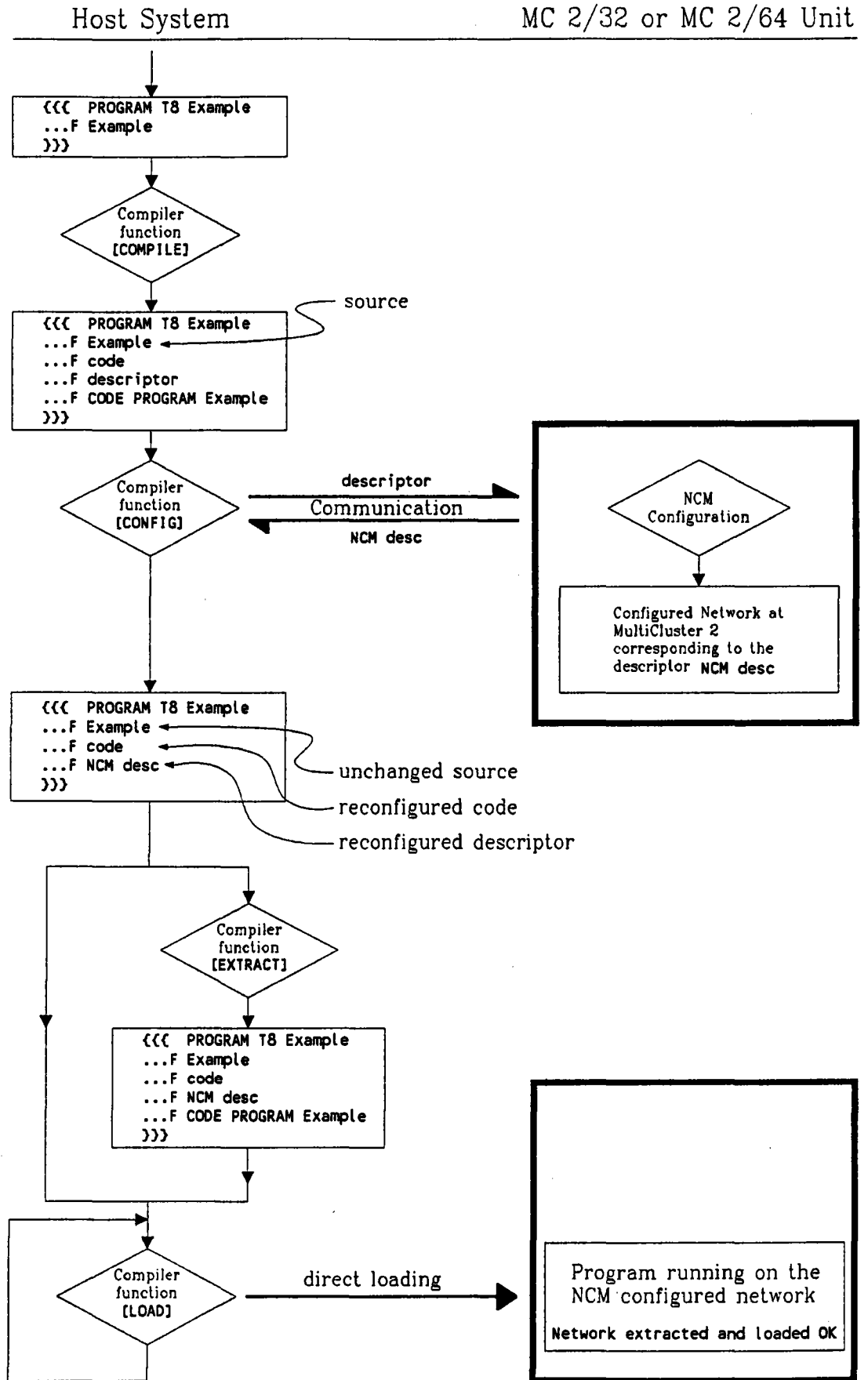


Fig 3.22 How to boot and run a program with the extended occam 2 compiler utilities.

Operation of the Function [CONFIG]

The following explanations will go through all steps which are necessary to achieve a running program on your MultiCluster 2/32 or /64. Figure 3.22 will illustrate the changes which will be made with the PROGRAM foldset during this process.

Let us assume you have an occam PROGRAM foldset containing the source fold with the name "Example".

After compiling this foldset with the function **[COMPILE]**, it will contain the source fold, the code fold, the descriptor fold and the extracted code fold in this order. Note that this function will execute in the usual way.

Now a new step compared to conventional operation is inserted. We execute the **[CONFIG]** function of the compiler utilities. This will force the compiler to analyze the present descriptor fold and send the network information to the NCM via the specified link (link specified in the toolkit fold "Transputer config parameters").

The NCM receives the messages and data and tries to compute a link mapping for the desired network. If this succeeds and the requested resources are available, the appropriate links are switched by programming the crossbar switches.

Next the NCM sends back the network structure with the new link numbering. The compiler receives these data and recompiles the program with the modified link numbers and generates a **new descriptor fold with the name »NCM desc«**. This descriptor contains the network structure of the network which is realisable on the MultiCluster 2.

Thus, the existence of an »NCM desc« fold is also an indicator for you to recognize whether a configuration attempt has previously been made or not.

Now a transputer network is available which is compatible to a manually connected network. Note that the configuration process described above will neither change the program sources nor the structure of your network.

The third step for you to process is to execute the **[LOAD]** function of the compiler utilities. This function is processed in the usual way without any modifications. There is no communication between the NCM and the compiler during this loading which is absolutely conventional.

Note : Do not forget to check the settings of the "Transputer loader parameters" in the toolkit fold ! In particular the load link number has to be the same as the config link number.

Hint : If you are interested in the new link numbering, let the compiler function [COMPILATION INFO] work your program and look at the result in the fold »CONFIG INFO PROGRAM« : »Wiring Diagram«. Another possibility is to use the »EXE Config and Boot« function <d>.

Warning : Don't use link placements within in SC folds, because for these NO renumbering is done !

Transputer Config Parameters

As mentioned above the behaviour of the function [CONFIG] of the compiler utilities is controlled by the parameter settings of the fold "Transputer config parameters":

```
{{{ Transputer config parameters
VAL link          IS 2 : -- 1, 2, 3
VAL compile      IS TRUE :
}}}
```

The parameter 'link' controls which link is used by the compiler to send a configuration request to the NCM, the so called config link.

The parameter 'compile' controls whether the compiler is forced to compile the program when invoking the function [CONFIG].

If you execute the function [CONFIG], the compiler looks for a valid »NCM desc« fold within the foldset. If this fold is found and the 'compile' parameter is FALSE, the compiler sends this reconfigured descriptor information to the NCM and forces it to realise this network without reconfiguration.

The compiler assumes that this previously computed descriptor fold is suitable and does not recompile the program.

If the normal descriptor fold is found inside a valid foldset, a reconfiguration with recompiling is started.

If no descriptor fold is found or the 'compile' parameter is TRUE, the whole configuration process is performed as described above with compiling, reconfiguration and recompiling.

Note : The MultiCluster 2/64-4 NCU is a clustered machine and, thus, may distribute the processors of a requested network on both clusters. During multiuser operation this may result in non-deterministic distributions. Therefore, old »NCM desc« folds can not be used to configure a network again, i.e. for the MultiCluster 2/64-4 NCU you have to set the 'compile' parameter **always** to TRUE.

This will force the NCM to compute a new configuration for the network and the compiler to recompile the program each time you invoke the [CONFIG] function.

Error and Warning Messages

Messages from the occam 2 compiler utilities are printed, as usually, in the first line of the screen. Also messages from the NCM are displayed in this way.

The following messages may occur:

```
[CONFIG] -> 'fold type must be a compiled program'
```

The syntax check of the compiler has detected an error. The cursor is placed at a position inside the source fold. Recompile with [CONFIG] is not possible in this case.

Use [COMPILE] to get an explicit error description of the detected error.

```
[CONFIG] -> 'Network Configuration Manager does not  
respond'
```

Possible reasons:

- no physical link connection to the user entry of the MultiCluster 2. If this link connection exists, check the internal connections of the user entry plugs in the MultiCluster 2 (see Technical Documentation).
- check the setting of the config link in the 'Transputer config parameter' fold.
- another user configures his network at that moment. Try again !
- user access is implicitly locked due to dispatcher activities.
- system software has not yet been loaded by dispatcher.

```
[CONFIG] -> 'partitioning failed - result = -804'
```

The number printed as result is a NCM error code described in Appendix C "Error Numbers". The error code -804, for example, means that a requested special processor is not available.

Another failure (`result = -1501`) may occur on clustered machines if the NCM is not able to find a correct partition for the network. This may happen if the structure cannot be mapped into the clusters because the actual distribution of free processors is inappropriate for this network.

A failure with `result = -301` occurs if no free link exists for the first processor in the network to get a connection to the host (cf. section 3.5 "User Entries").

Check the program fold for all requirements given in sections 3.3-3.5 and also check the connections of the input link and the special processors.

[CONFIG] -> 'not enough processors for new configuration'

If the total number of processors in the system is exceeded, this error message is produced. Note that the processors are distributed among the connected users and, thus, not all processors may be free. The present state of the system can be displayed by using the <s> function of the »EXE Config and Boot network«.

[LOAD] -> 'failed to output boot code ...'

There might be several reasons for this message:

- no physical link connection between the host transputer and the network.
- the config link and the load link differ from each other. Check the folds 'Transputer config parameters' and 'Transputer loader parameters'.
- a wrong load link selection.
- attempt to load another program onto the previously configured network.

3.1.4 Example PROGRAMs for MultiTool

To learn from examples how the logical processor numbers are specified in occam 2 PROGRAMs, the following two configuration sections are shown.

Example 1 :

You want to install a pipe with length n as your network. You can use the occam PLACED PAR like this :

```

PLACED PAR
  PROCESSOR 1000 T8
    PLACE to.host      AT 1:
    PLACE frompipe [0] AT 4:
    SC.start.of.pipe (to.host, frompipe [0] )
  PLACED PAR i = 0 FOR n
    PROCESSOR i T8
      PLACE frompipe [i]      AT 0:
      PLACE frompipe [(i+1)] AT 6:
      SC.in.pipe(frompipe [i], frompipe[(i+1)] )

```

All processor numbers are between 0 and 1023. The Processor 1000 is the first declared processor in the PLACED PAR with a valid MultiCluster 2 / SuperCluster processor number.

The first link not connected within the MultiCluster 2 / SuperCluster is link number 1. Hence the NCM connects link 1 of processor 1000 with the host (via the user entry (first input link)).

Example 2 :

You want to use the GDS board found in the first slot for special processors.

```

PLACED PAR
  PROCESSOR 1100 T8 -- GDS
    PLACE to.host      AT 3:
    PLACE from.node [0] AT 4:
    PLACE from.node [1] AT 5:
    PLACE from.node [2] AT 6:
    SC.GDS.process (to.host, from.node [0],
      from.node [1], from.node [2])
  PLACED PAR i = 0 FOR 3
    PROCESSOR i T8
      PLACE from.node [i] AT 0:
      SC.GDS.slave (from.node [i])

```

The processor 1100 is the first processor in the system. Its first free link is link 3 which is connected with the host.

All other processors are connected to the GDS.

3.2 USER INTERFACE SOFTWARE FOR HELIOS

The user interface software for Helios is represented by the Helios program »hconfig« (Helios Network Configuration for MultiCluster 2 and SuperCluster) and several example resource maps.

If you want to run the Helios Operating System on an electronically reconfigurable machine, you first have to configure the physical network for Helios.

This configuration is done by the Network Configuration Manager (NCM).

To communicate to the NCM from Helios you need the Helios Network Configuration Support Utility »hconfig«.

The program »hconfig« produces the input for the NCM from the binary Helios resource map and sends this configuration request to the NCM. (The binary resource map is produced by the Helios command `rmgen` from the resource map edited by the user.)

The »hconfig« may also receive a renumbering list from the NCM and build a new resource map with renumbered links (for MultiCluster 2/32 and /64).

Different versions

Since version 3.0, there is one common »hconfig« for MultiCluster 2 and SuperCluster.

Former versions have been divided into a MultiCluster 2 and a SuperCluster version.

The only difference was in computing the user id and channel numbers for user information. Therefore, you may as well use the MultiCluster version to configure the SuperCluster and vice versa.

Note, however, that the »hconfig« version 2.0 **does not** support any special processors.

Preview on Helios 1.2

User of Helios 1.2 may use »hconfig« in connection with the Helios 1.1 `rmgen`. Note that for use of the *new* `rmgen`, there might be necessary a software update.

3.2.1 Hardware Connection to the MultiCluster 2 / SuperCluster

Connect e.g. link 2 of your host transputer to the MultiCluster 2 / SuperCluster Link Interface (user entries). The wiring scheme is described in the Technical Documentation.

The **MultiCluster 2** has only **one** user entry whereas the **SuperCluster** has **two** different user entries (cf. section 3.5 "User Entries").

3.2.2 Manual Usage of Helios »hconfig«

If you have booted Helios at least on one node, you are able to apply the user interface software simply by typing the command »hconfig«.

The syntax of this »hconfig« is as follows:

```
hconfig [-m] [-l] [-s] [-f] [-o] [-yes] [-r]
        [-p] [-h] [-i] [-a]
```

Options:

-m <file_name> The **.map** file for the binary resource map used to configure a network can be given to »hconfig« via this option.

If **-m** option is missing, a default .map file (call **-i** option) is used.

-l <link_number> **Link** of the host processor connected to a user entry of a MultiCluster 2 or SuperCluster.

If **-l** option is missing, a default link (call **-i** option) is used.

-s **Second Configuration** or **SuperCluster Configuration.**

The resource map is sent to the NCM. The NCM configures exactly the given network. (Cannot be used for MultiCluster 2/64-4 NCU.)

If the network cannot be realised (e.g. on a MultiCluster 2), the resource map should be modified first by **-f** option.

- f** [**<file name>**] **First Configuration for MultiCluster 2/32 and /64.** The NCM configures a network which can be realised on a MultiCluster 2 and returns a link mapping to »hconfig« which generates a new textual resource map usable for **-s** option.
- <file name>** is the textual input file used to generate the new textual resource map. It should correspond to the binary resource map given by option **-m** or default.
The syntax of the textual input file will be given below.
- o** **<file name>** Textual **output file** with a NCM generated and modified resource map corresponding to the network configured by the NCM. (MultiCluster 2 option)
- yes** Delete an existing file given via option **-o**.
Option **-a** becomes 'non interactive'.
- r** **Return network to NCM :**
All processors once requested (at the same user channel of the machine) are returned to the processor pool of the NCM.
- a** **Show system status** of MultiCluster 2 or SuperCluster. The number of free processors and the usage of processors are printed to standard output. This option also shows the internal placement of your network within the system.
- Do not use** this option if Helios is already booted at the configured network. It will reset the processor connected to the used link.
Using option **-yes**, this option becomes usable for batch; it **disables** the interactive mode.
- p** The processors and links defined in the resource map are **printed** to standard output. (If **-s** or **-f** option is used after **-p** option, the output is also send to the NCM.)
- h** Online help.
- i** Information about defaults and others.

Some examples

```
hconfig -s -l 2 -m /helios/etc/xxx.map
```

sends a configuration request based on the resource map »xxx.map« to the Network Configuration Manager via link 2. Link 2 is the link connected to the configured network afterwards.

```
hconfig -r
```

returns all requested processors to the processor pool of the NCM for further use by other users. It should be called any time before you logout from Helios.

```
hconfig -m default.map -f default.rm -o ncm_mod.rm
```

sends a configuration request based on the resource map »default.map« to the NCM via the default link (cf. `hconfig -i`).

The NCM computes a link mapping and realises a modified network on the MultiCluster2. The »hconfig« receives the link mapping information and generates a modified textual resource map, if the file does not already exist.

Otherwise you have to use `-yes` to allow »hconfig« to **overwrite** the existing file. This textual resource map may be treated as usual.

```
hconfig -l 1 -a -yes > status.inf
```

The momentary configuration status is received from the NCM via link 1, and sent into the file »status.inf« at the actual directory.

The option `-yes` **disables** interactive mode of option `-a`.

The two following figures summarise all possible actions on the way to a Helios network on a MultiCluster 2 or SuperCluster respectively. The second figure should help to illustrate the more complex configuration procedure on the MultiCluster 2/32 and /64 units.

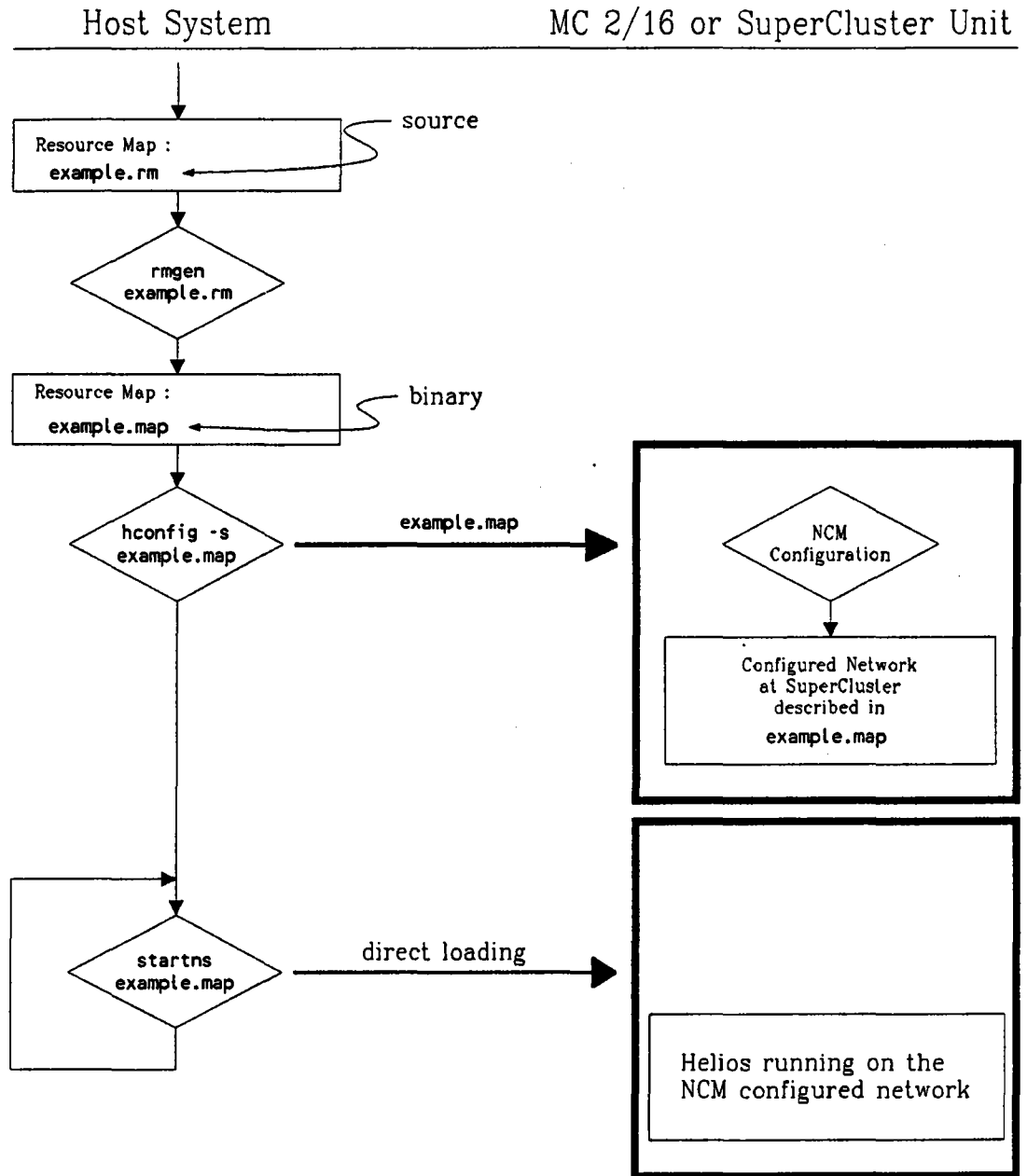


Fig 3.23 How to run Helios on a MultiCluster 2/16-1 NCU or SuperCluster.

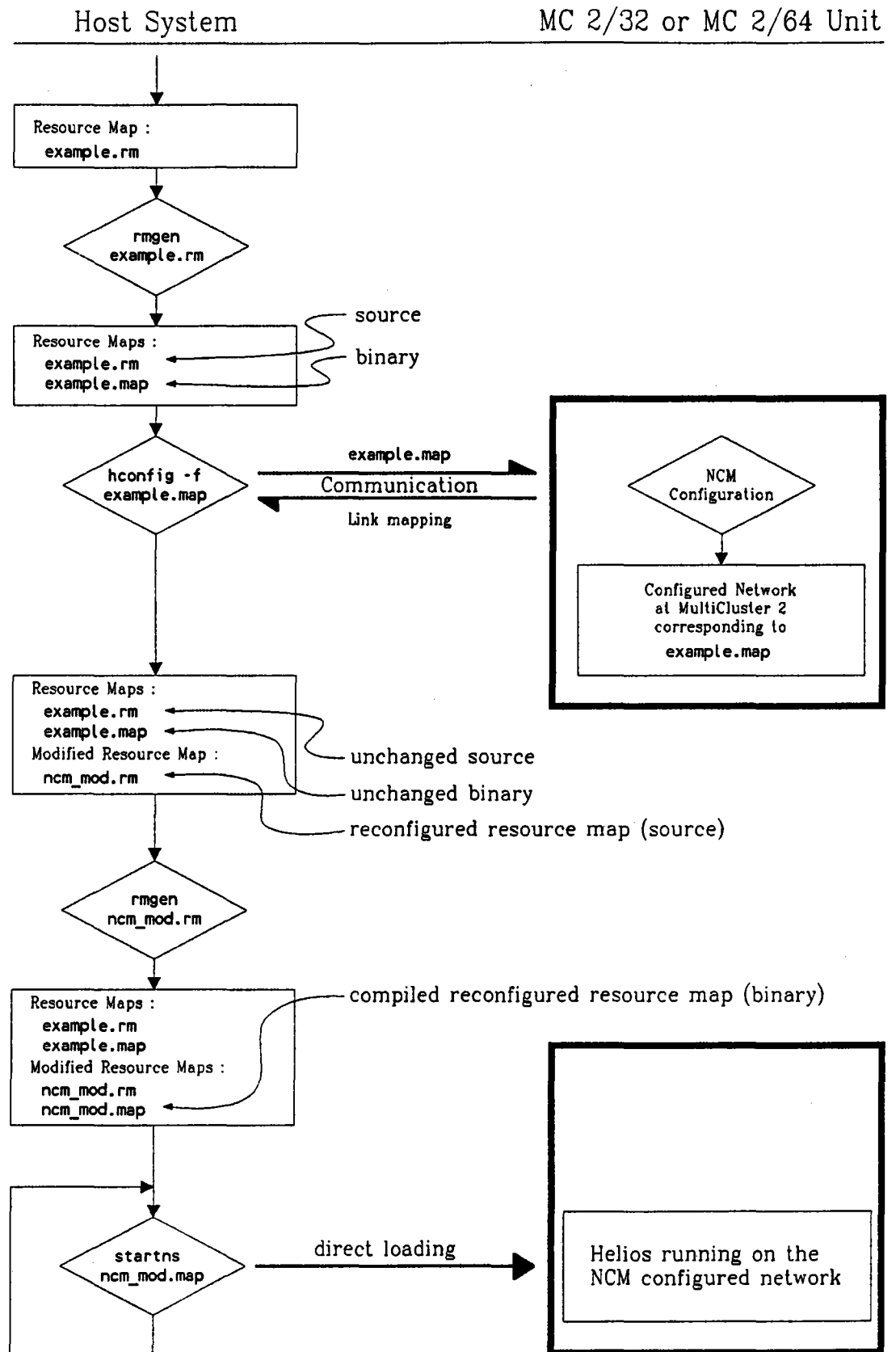


Fig 3.24 How to run Helios on a MultiCluster 2/32-2 NCU or MultiCluster 2/64-4 NCU.

3.2.3 Automatic Usage of »hconfig« on PC Hosted Systems

To configure a network connected to link 2 (default link) of your host transputer the first two respectively three lines of your `etc/initrc` should look as follows:

For MultiCluster 2/16-1 NCU and SuperCluster

```
run -e -w /helios/bin/hconfig hconfig -s -m /helios/etc/default.map
```

```
run -e /helios/bin/startns startns /helios/etc/default.map
```

For MultiCluster 2/32-2 NCU and MultiCluster 2/64-4 NCU

```
run -e -w /helios/bin/hconfig hconfig -m /helios/etc/default.map  
-f /helios/etc/default.rm -o /helios/etc/ncm_mod.rm -yes
```

```
run -e -w /helios/bin/rmgen rmgen -o /helios/etc/ncm_mod.map /helios/etc/ncm_mod.rm
```

```
run -e /helios/bin/startns startns /helios/etc/ncm_mod.map
```

If your file `etc/initrc` does not look like above, edit the file and insert the lines on the top.

Please remove option `-r` from »startns«. Without this option the »startns« command resets the processor network before booting.

Important: Please add to your logout script the following line. This makes your processors available for other users after logging out.

```
hconfig -r
```

The Helios boot procedure is now as follows in a single step:

- Helios is booted on the root transputer.
- The network is allocated and configured.
- The network server is started.
- The task force manager is started.

Logging out returns the used network to the NCM.

3.2.4 Automatic Usage of »hconfig« on SUN Hosted Systems

Edit your private `loginrc` file located in your home directory, e.g. `/helios/usr/username`.

Suppose you have created a subdirectory »conf« in your homedirectory to hold your different resource maps.

To configure a network connected to link2 (default link) of your host transputer, add to your `loginrc` file:

For MultiCluster 2/16-1 NCU and SuperCluster

```
hconfig -s -m conf/default.map
```

```
startns conf/default.map
```

For MultiCluster 2/32-2 NCU and MultiCluster 2/64-4 NCU

```
hconfig -m conf/default.map -f conf/default.rm -o conf/ncm_mod.rm -yes
```

```
rmgen -o conf/ncm_mod.map conf/ncm_mod.rm
```

```
startns conf/ncm_mod.map
```

Please remove option `-r` from »startns«. Without this option the »startns« command resets the processor network before booting.

Important: Please add to your logout script the following line. This makes your processors available for other users after logging out.

```
hconfig -r
```

The Helios boot procedure is now as follows in two steps:

1. Booting the root transputer.

Login ...

2. Execution of the »loginrc« file:

- The network is allocated and configured.
- The network server is started.
- The task force manager is started.

Logging out returns the used network to the NCM.

3.2.5 Setting Up the Resource Map

If you have installed »hconfig« in your Helios and done the installations for the automatic usage of »hconfig«, you need to copy the binary resource map file for the desired network to »default.map« (located in your directory for resource maps) to get a special configuration.

For MultiCluster 2/32-2 and /64-4 NCU please copy also the textual resource map to »default.rm«.

Type:

```
cd etc                for PC hosted systems
cd conf              for SUN hosted systems
cp xxx.map default.map
cp xxx.rm default.rm  for MultiCluster 2/32-2, 64-4 NCU only
cd ..
```

at Helios level, where `xxx` is your desired resource map.

Be sure that the Network Configuration Manager is running on the MultiCluster 2 / SuperCluster.

Now simply boot Helios.

3.2.6 Syntax of the Resource Map (MC 2/32 and /64)

The following paragraph explains the available and significant **syntax** according the `.rm` file

Significant input lines:

```
terminal <proc_id> {<link(0)>,<link(1)>,  
                  <link(2)>,<link(3)>;<rest_text>
```

terminal	<p>»hconfig« accepts only two possible spellings of the keyword terminal, namely TERMINAL or terminal. Any other spelling causes »hconfig« to terminate.</p> <p>Only one terminal is allowed in each line, except for comment.</p>
<proc_id>	<p>The processor identification consisting of any characters except the <code>{</code>.</p> <p>Note, however, that »hconfig« only works on processors identified with numbers in the interval displayed by the <code>-i</code> option.</p> <p>Lines with other numbers are regarded as external processors. The line is copied 1:1 to the output file.</p>
{	Significant sign for start of link connection table.
<link(i)>	<p>Any text except for <code>"</code> or <code>;"</code> describing your intended connection of link(i), where $i = 0..3$. Example: <code>link0 = ^02</code></p>
,	<p>Separator for the link texts <code><link(0)> - <link(3)></code>.</p>
;	End of significant line.
<rest_text>	A single carriage return or arbitrary text without keyword.

All the significant signs and expressions have to be in **one single line** since the keyword has occurred (no carriage return may appear in the significant input line).

Only white spaces (tabs, spaces or nothing) may occur before the **keyword**, between **keyword** and **<proc_id>** and between **<proc_id>** and **{**.

You may insert **comment** by using **"--"** as significant sign, then the rest of the line will be skipped.

Every **line** may contain up to 256 characters.

Generated output:

Each **significant input line** is changed to the following single output line inside the produced output file given by **-o** option:

```
terminal <proc_id> {<A(0)>,<A(1)>,<A(2)>,<A(3)>;
             <rest_text>
```

<proc_id>	Same as in input file.
<A(j)>	Text <link(i)> of the significant input line, where the logical link <i>i</i> was mapped to the physical link <i>j</i> by the NCM, where $i, j = 0..3$. Let <i>f</i> be the mapping of logical to physical links <i>f</i> : link <i>i</i> → link <i>j</i> and <i>g</i> be the inverse mapping (physical to logical link) <i>g</i> : link <i>j</i> → link <i>i</i> then <i>A(j)</i> gives the text for link position <i>j</i> : <A(j)> = <link(g(j))>
<rest_text>	Unchanged <rest_text> of significant input line.

Non-significant lines and **comments** are copied to the same relative position according to the input file.

Layout of the new resource map:

»hconfig« only uses one spelling of the **keyword terminal**, that is in lower letters.

Writing the new resource map, »hconfig« inserts the **keyword** on the first tab-position, so that **terminal** in the output file is always shifted one tab to the right.

White spaces between the keyword, <proc_id> and { are changed to one blank.

Hint:

You have two possibilities to control the reconfiguration:

- either set all **keywords** in the input file to the beginning of each line; now you recognize the reconfigured lines by the changed position of **terminal**.
- or double every line and mark one of them as comment; now you can compare them directly.

3.2.7 Some Useful Hints

If the last Helios user finishes his session, he should call »hconfig -r« before logging out, to return reserved processors to the processor pool of the Network Configuration Manager, which may distribute these processors to other users.

Be careful with sending configuration or status requests to the NCM, this will disturb your present network configuration on which your Helios is running.

If you edit a Helios resource map, all nodes (terminal nodes) which have a number, e.g. between 1 and 1999 behind the last '/' (for the actual range of numbers call »hconfig -i«) are regarded as MultiCluster 2 / SuperCluster nodes by »hconfig«. The NCM then separates the relevant processor numbers for the configuration (cf. next chapter). The node of the host processor should have the number /00.

Note the warning in chapter 3.5 "User Entries". The network configured by the NCM may differ from your resource map.

3.2.8 Error Messages

All errors produced by »hconfig« generate a textual output on standard error.

Nevertheless, some errors shall be analysed here:

error description :

```
hconfig version 3.0 (901204) hbb + cr
Copyright (C) 1989,1990, Parsytec GmbH
```

```
mapfile      == /helios/etc/default.map
link to NCM  == 2
```

The Network Configuration Manager is not responding.

Check the following:

Have you got a link connected to the machine's user entry ?

May be another user is just configuring a network.

Is the NCM already started up ?

result : The NCM could not be ringed up, no configuration is done.

recovery : Several reasons for this failure are possible:

1. The wrong link number was specified, cf. -l option of »hconfig«.
Call `hconfig -l <link number> ...`
2. No link of the host transputer is connected to the user entry of the MultiCluster 2 / SuperCluster (cf. chapter 3.2.1 "Hardware Connection to the MultiCluster 2 / SuperCluster" and Technical Documentation).
Connect a host transputer link to a first user entry.
If you use a SUN with MTM-Sun board, the configuration on the MTM-Sun may be faulty.
3. Another user configures a network (a few seconds) or the dispatcher is operating via the system interface (interactive). Try again later.
4. No MultiCluster 2 or SuperCluster is available.
Remove »hconfig« call in your »initrc« or »loginrc« file by placing a # in front of the line.
5. The dispatcher did not load the Network Configuration Manager onto the MultiCluster 2 / SuperCluster (cf. chapter 2.2 "Running and Using the System Software").

error description :

```
hconfig version 3.0 (901204) hbb + cr
Copyright (C) 1989,1990, Parsytec GmbH
```

```
mapfile      == /helios/etc/default.map
link to NCM == 2
```

	Your User Id.	Network Channels
SuperCluster :	6	74 and 75
MultiCluster 2 :	11	74

```
Partitioning time : 0.0 seconds
Network configured ok.
```

```
ns : Network server version 2.02.
pa_rboot on /Cluster/00 for /Cluster/1100 : failed to
                                         send bootstrap size.
/Cluster/ns : Error while booting /Cluster/1100
/Cluster/ns : The link between /Cluster/1100 and
              /Cluster/00 may be faulty
```

result : A network has been configured by the NCM, but Helios »startns« is not able to boot this network.

recovery : Several reasons for this failure are possible:

1. Ensure that »hconfig« said: 'Network configured ok.' otherwise »startns« should not be able to boot.
2. You have used the Helios loader »startns« with the option `-r`. With this option »startns« expects a processor network already reset. Remove the option `-r` from your »startns« call to do a reset for every processor inside the MultiCluster 2 / SuperCluster network.
3. You may have ignored the special requirements for the resource map, that always the **first processor** with a MultiCluster 2 / SuperCluster processor number (cf. section 3.3 "Special Requirements for Processor Numbering") is connected via its first free link (link not used inside the MultiCluster 2 / SuperCluster) to the root transputer. Read chapter 3.5 "User Entries" and rewrite your resource map.
4. Only SuperCluster: You have used a second user entry to configure your network. The network is always connected to the first user entry. Use the corresponding first user entry and boot again.

error description :

```
hconfig version 3.0 (901204) hbb + cr
Copyright (C) 1989,1990, Parsytec GmbH
```

```
mapfile      == /helios/etc/default.map
link to NCM == 2
```

	Your User Id.	Network Channels
SuperCluster :	5	72 and 73
MultiCluster 2 :	9	72

```
Partitioning time : 0.0 seconds
```

```
NCM Error Code : -804
```

```
Failed : Special processor is not free.
```

result : The NCM itself detected some fact so that a configuration is not possible (here: a special processor is not free).

recovery : The error code (here: -804) is described in Appendix C. The special recovery is given there for each error code.

3.2.9 Exit Codes

»hconfig« may terminate with the following exit codes (since version 3.0):

Succeed:

0 : no error

File:

1 : error opening .map file
2 : error calling ftell
3 : error calling fseek
4 : cannot allocate memory for file-buffer
5 : error reading .map file
6 : error while closing file
7 : bad command line (syntax error)
8 : failed to extract information from .map file
9 : error writing file
10 : .rm file already exists
11 : .map file empty

Timeouts:

20 : timeout receiving from link in
21 : timeout while sending on link
22 : timeout receiving from link for first time

NCM:

30 : failed to partition network
31 : not enough processors available
32 : error partitioning network
33 : configuration failed
34 : failed to connect network
35 : special processor not available
36 : special processor not existent
37 : two processors defined with second user entry
38 : first processor defined has no free link
39 : no second configuration possible on MultiCluster 64

Link:

40 : cannot change mode of link
41 : cannot allocate link
42 : unable to free link
43 : cannot change link to previous mode
44 : ConfigIn link used for network request

Syntax:

50 : error while reading .rm file
51 : syntax error in .rm file
52 : line too long in .rm file

The exit code can be used in shell scripts (e.g. `loginrc`) to stop running other programs (e.g. `startns`) in case of an »hconfig« error.

3.2.10 Example Resource Maps for Helios

Example 1 (MultiCluster 2):

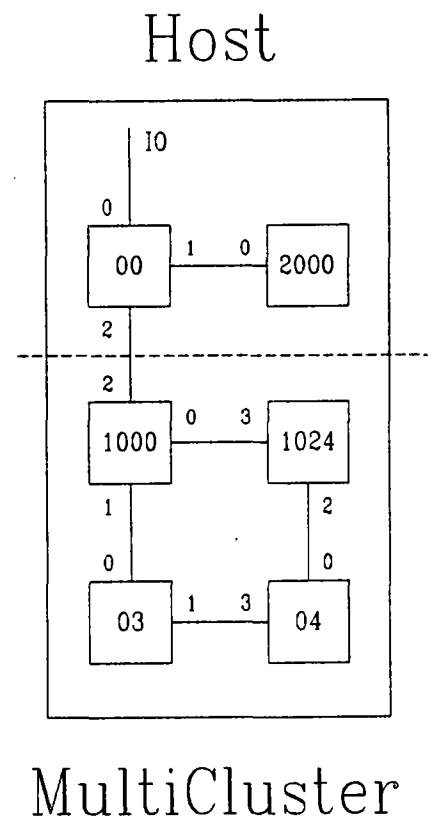
The following resource map realises a 2 x 2 matrix inside the **MultiCluster 2** connected via one node to the host transputer link 2. At link 1 of the host an **external** processor is connected.

The first node inside the MultiCluster 2 system, processor 1000, is connected via its link 2 to the host transputer. Link 0 and link 1 are connected inside the MultiCluster 2.

```

subnet /Cluster {
CONTROL Rst_An1 [/Cluster/00];
terminal 00    {~10,~2000,~1000, ; SYSTEM;
                Mnode Rst_An1(pa_ra.d);
                ptype T800;
            }
terminal 1000 {~1024,~03,~00, ; HELIOS;
                memory #100000;
                ptype T800;
            }
terminal 1024 { , ,~04,~1000; HELIOS;
                memory #100000;
                ptype T800;
            }
terminal 03   {~1000,~04, , ; HELIOS;
                memory #100000;
                ptype T800;
            }
terminal 04   {~1024, , ,~03; HELIOS;
                memory #100000;
                ptype T800;
            }
terminal 2000 {~00, , , ; HELIOS;
                memory #100000;
                ptype T800;
            }
terminal 10   { ; 10; }
}

```



On the left side the resource map is shown and on the right side the according chart of the resulting network.

The processor 00 is the **master node**. This number is not sent to the NCM by »hconfig« and, therefore, it is no processor inside the MultiCluster 2 system.

The **first processor** inside the system is processor 1000. It gets a connection via its **first free link** to the master node. The first free link in this example is link 2. This is a free link inside the network because the NCM does not know the processor 00.

Note that it is **not** possible to connect node 1000 via link 3 to the master node if link 2 of node 1000 remains free ! In this case you would get exactly the same network as in this example, but Helios would try to boot the network in the resource map.

There are four **MultiCluster 2 processors** (1000, 1024, 03 and 04) because they have numbers between 1 and 1024.

Processor 2000 is not inside the system because it is no valid number for the NCM (cf. chapter 3.3 "Special Requirements for Processor Numbering").

Example 2 (SuperCluster):

The following resource map realises a 2 x 2 matrix inside the SuperCluster connected via one node to the host transputer link 2. At link 1 of the host an external processor is connected.

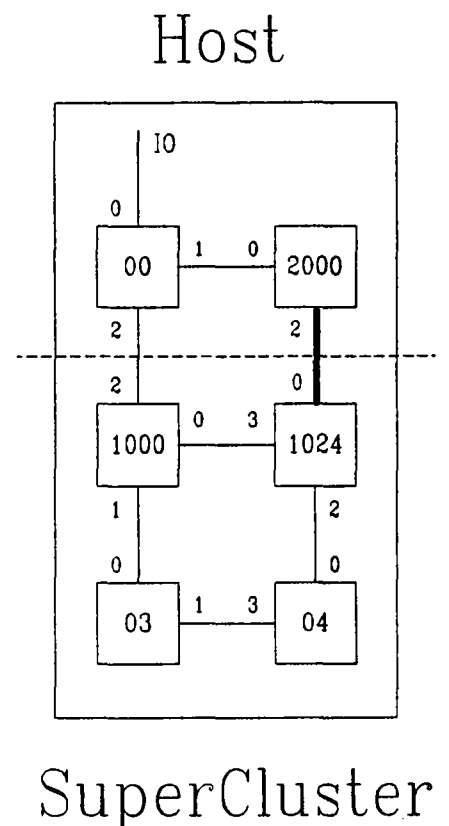
The first node inside the SuperCluster system is connected via its link 2 to the host transputer. Link 0 and link 1 are connected inside the matrix.

Note that in this example the **second user entry** (first free link of processor 1024) is connected to the **external processor 2000**.

```

subnet /Cluster {
CONTROL Rst_An1 [/Cluster/00];
terminal 00    {~10,~2000,~1000, ; SYSTEM;
                Mnode Rst_An1 [pa_ra.d];
                ptype T800;
            }
terminal 1000  {~1024,~03,~00, ; HELIOS;
                memory #100000;
                ptype T800;
            }
terminal 1024  {~2000, ,~04,~1000; HELIOS;
                memory #100000;
                ptype T800;
            }
terminal 03    {~1000,~04, , ; HELIOS;
                memory #100000;
                ptype T800;
            }
terminal 04    {~1024, , ,~03; HELIOS;
                memory #100000;
                ptype T800;
            }
terminal 2000  {~00, ,~1024, ; HELIOS;
                memory #100000;
                ptype T800;
            }
terminal 10    { ; 10; }
}

```



On the left side the resource map is shown and on the right side the according chart of the resulting network.

Processor 00 is the **master node**. This number is not sent to the NCM by »hconfig« and, therefore, it is no processor inside the SuperCluster system.

The **first processor** inside the system is processor 1000. It gets a connection via its **first free link** to the master node. The first free link in this example is link 2. This is a free link inside the network because the NCM does not know the processor 00.

Note that it is **not possible** to connect node 1000 via link 3 to the master node, if link 2 of node 1000 remains free ! In this case you would get exactly the same network as in this example, but Helios would try to boot the network in the resource map.

There are four **SuperCluster processors** (1000,1024, 03 and 04) because they have numbers between 1 and 1024. Note the special meaning of processor 1024 !

Processor 2000 is not inside the system because it is no valid number for the NCM, although it is connected to the network via **second user entry** (only possible on SuperCluster).

3.3 SPECIAL REQUIREMENTS FOR PROCESSOR NUMBERING

The Network Configuration Manager needs some special information about the network to be established on the MultiCluster 2 / SuperCluster. The NCM gets this information from the numbering and ordering of the processors in the program fold or the resource map respectively.

The following numbers for processors given by the user are interpreted by the NCM as processors located in the MultiCluster 2 / SuperCluster:

0	<p>MultiTool: normal processor inside the MultiCluster 2 / SuperCluster</p> <p>Helios: special Helios node which is not located in the MultiCluster 2 / SuperCluster (The NCM would accept this number but the »hconfig« will not send it to the NCM).</p>
1 ... 1023	normal processors inside the MultiCluster 2 / SuperCluster
1024	<p><u>if</u> a second user entry exists, the processor 1024 gets a connection to it via its first free link (SuperCluster)</p> <p><u>else</u> a normal processor as 0-1023 (MultiCluster 2)</p>
1100, 1200, ... [*] 1200	<p>special processors located at fixed NCU channels.</p> <p>The relation between these logical numbers for special processors and the assigned NCU channels is shown in the tables below.</p> <p>The numbers of available special processors may be displayed with the status request of the »EXE Config and Boot network« or by »hconfig a« before a user has configured a network.</p>
1124, 1224, ... [*]	<p><u>if</u> a second user entry is available: special processors with second input (for SuperCluster only):</p> <p>1124 : like processor 1100 but with the first free link connected to the second user entry.</p> <p>1224 : like processor 1200 ...</p>

* The »hconfig« (user interface software for Helios) version 2.0 does not support the numbers for special processors.

.
. etc.

(cf. 1024 and 1100, 1200, ...)

else these numbers are skipped by the NCM as external processors (MultiCluster 2)

first valid number The first occurrence of one of the above numbers in a resource map or a descriptor fold is interpreted as the processor connected to the (first) user entry. The first free link is used to be connected to the user entry.

all other numbers are skipped by the NCM as external processors, which need not be connected via the NCM. These processors are not located inside the MultiCluster 2 / SuperCluster.

text **Helios:** If text occurs in a Helios resource map to denote a processor, the »hconfig« handles this processor as external node, i.e. it does not send any number to the NCM.

MultiTool: not possible.

3.4 CONNECTING OF SPECIAL PROCESSOR BOARDS

If you have transputer boards with special processors integrated into your system, the links of these special processors have to be connected with some NCU channels depending on the type of system.

The technical documentation will help you to find the plugs of these channels.

The following tables show which logical processor numbers are reserved to allocate the special processors connected to the particular NCU channels.

The NCU numbers are physical numbers given in Appendix D "Physical Device Numbers".

o MultiCluster 2/16-1 NCU

proc number	NCU	NCU channel	proc links
1100	0	92-95	0-3
1200	0	88-91	0-3
1300	0	84-87	0-3
1400	0	80-83	0-3
1600	0	76-79	0-3
1700	0	72-75	0-3

o X MultiCluster 2/32-2 NCU

proc nr	link 0 NCU:chan	link 1 NCU:chan	link 2 NCU:chan	link 3 NCU:chan
1100	0:72	1:72	0:73	1:73
1200	0:74	1:74	0:75	1:75
1300	0:76	1:76	0:77	1:77
1400	0:78	1:78	0:79	1:79
1500	0:80	1:80	0:81	1:81
1600	0:82	1:82	0:83	1:83
1700*	0:84	1:84	0:85	1:85
1800*	0:86	1:86	0:87	1:87

* available since Version 3.0

o MultiCluster 2/64-4 NCU

proc nr	link 0 NCU:chan	link 1 NCU:chan	link 2 NCU:chan	link 3 NCU:chan
1100	2:64	3:64	2:65	3:65
1200	2:66	3:66	2:67	3:67
1300	2:68	3:68	2:69	3:69
1400	2:70	3:70	2:71	3:71
1500	2:72	3:72	2:73	3:73
1600	2:74	3:74	2:75	3:75
1700	0:72	1:72	0:73	1:73
1800	0:74	1:74	0:75	1:75

o SuperCluster 16, 32

proc number	NCU	NCU channel	proc links
1100	0	92-95	0-3
1200	0	88-91	0-3
1300	0	84-87	0-3
1400	0	80-83	0-3

o SuperCluster 48, 64

proc number	NCU	NCU channel	proc links
1100	0	92-95	0-3
1200	0	88-91	0-3
1300	0	84-87	0-3
1400	0	80-83	0-3
1500	5	76-79	0-3
1600	5	72-75	0-3
1700	5	68-71	0-3
1800	5	64-67	0-3

o SuperCluster 128

proc number	NCU	NCU channel	proc links
1100	5	64-67	0-3
1200	5	68-71	0-3
1300	5	72-75	0-3
1400	5	76-79	0-3
1500	6	80-83	0-3
1600	6	84-87	0-3
1700	6	88-91	0-3
1800	6	92-95	0-3

3.5 USER ENTRIES

In each machine 8 users are administrated. A single user has access to the machine via one link on a MultiCluster 2 and via two links on a SuperCluster.

These links are used to ring up the NCM and to get a connection to the configured network afterwards.

Take care that ringing the NCM resets the first processor in the network connected to the user entry.

3.5.1 Processor Connected to the First User Entry

Due to historical reasons the NCM is not yet able to receive information about the processor and its link connected to the host processor outside the system. (The descriptor fold of an occam program does not contain this information !)

Hence the NCM works as follows:

- It takes the *first processor* with a valid MultiCluster 2 / SuperCluster processor number (cf. section 3.3 "Special Requirements for Processor Numbering")
- The *first free link* of this processor (i.e. a link not used within the MultiCluster 2 / SuperCluster network) is connected to the (first) user entry (host).

Take care that always the first user entry is connected even if the configuration request is done via the second user entry (on a SuperCluster).

WARNING (for Helios only)

The configured network may differ from the *Helios* resource map if the user ignores the fact above !

3.5.2 The Second User Entry (for SuperCluster only)

The second user entry can be used to ring up the NCM in the same way as the first user entry.

Some difference between the first and second user entry is in configuring networks:

- The first user entry is always used and connected to the first valid SuperCluster processor.
- The second user entry is only connected if a certain processor number (e.g. 1024, 1124, ...) is used even if the configuration request is done via the second user entry.

Usage as Second External Link

A valid NCM processor number > 1000 with the last two digits being ..24 is regarded as processor with a second external link (only one such processor may occur in a configuration request).

The first free link (not used in the internal MultiCluster 2 / SuperCluster network) of this processor is connected to the second user entry.

Note that the same processor may be connected to the first and second user entry when the processor number for the second user entry is the first valid processor number in the resource map or descriptor fold.

Usage for Status Request

If the second user entry is not used as second external link in the configured network, you may use it for status requests while you are running a network on the machine without disturbing the network, because the reset signal sent to the NCM for ringing does not reach the application processors within the network.

3.5.3 Available User Entries

MultiCluster 2/16, /32, /64:

user id	NCU	NCU channel 1. user entry	NCU channel 2. user entry
1	0	64	-
2	0	65	-
3	0	66	-
4	0	67	-
5	0	68	-
6	0	69	-
7	0	70	-
8	0	71	-

SuperCluster 16, 32, 48, 64, 128:

user id	NCU	NCU channel 1. user entry	NCU channel 2. user entry
1	0	64	65
2	0	66	67
3	0	68	69
4	0	70	71
5	0	72	73
6	0	74	75
7	0	76	77
8	0	78	79

3.6 NCM USER INTERFACE PROTOCOL

The given protocols are based on the NCM version 3.0 (and earlier versions). They could be changed fundamentally in the next release !

To get user access via the NCM user interface from different host machines or operating systems two steps have to be implemented:

- Ringing the NCM
- Sending a request

3.6.1 Ringing the NCM

The hardware used (link adapter) should be able to pull down the reset and communicate via the same link at one time.

The steps of ringing are

1. Assert the reset signal on the ringing link.
2. Receive 15 bytes from the ringing link.
3. Send the received 15 bytes back to the NCM via the ringing link (synchronisation).
4. Receive the number of the channel to which the ringing link is connected. The channel number is an integer with 32 bits.

From the received channel number the user id may be determined from the lists "available user entries" in the chapter above ("User Entries").

5. Deassert the reset signal.

After ringing, the NCU main processor is connected to your ringing link and you are able to communicate to the NCM via the ringing link until the request is finished or the NCM got a timeout.

3.6.2 Sending a Request

First a function code (a 32 bit integer) is sent to the NCM.

Function Code	Function
0	return network to NCM
1	request a network configuration (without link renumbering)
3	status request
5	request a network configuration with link renumbering (by the Euler Method)
7	user data request

Function Code 0:

No further protocol is necessary. The user network is returned to the NCM.

Function Code 1:

The user network is described in a **descriptor** which is sent to the NCM after the function code 1.

The syntax of the descriptor will be given below (section 3.6.3 "Syntax of Network Descriptor").

The descriptor protocol is finished by sending line length 0.

Afterwards, the NCM sends a **return sequence** depending on the result:

receive **availability code AC** (32 bit integer) (don't use timeout):

AC = 0 : enough processors are available

AC <> 0 : AC - 1 is the number of available processors

IF

AC <> 0

receive number of requested processors (integer 32), protocol finish

AC = 0

receive the NCM **Error Code EC** (32 bit integer):

EC = 0 : network configured ok

EC < 0 : error configuring the network (error codes are described in Appendix C of this documentation); protocol finished

EC > 0 : not used

IF

EC = 0

receive **number of NCUs** (32 bit integer)

For each NCU:

- receive **NCU number** (32 bit integer)

- receive **96 integers 32 bit** describing the NCU connection for the channels 0-95.

Function Code 5:

The protocol of function code 5 is the same as for function code 1 except for the last part if EC = 0.

This part of the protocol is as follows:

IF

EC = 0

receive **number.of.six-pairs** (32 bit integer)

i = 0 FOR **number.of.six-pairs**

receive a single six-pair of 32 bit integer numbers (a,b,c,d,e,f)

where

a: current processor number generated by the user interface software (cf. first number after PROC keyword in descriptor syntax, sections 3.6.3)

b: logical processor number (denotation in the user program, cf. second number after PROC keyword in descriptor syntax, section 3.6.3)

link mapping:

c: the logical link 0 (link 0 of the user program) is mapped to the physical link c switched by the NCM)

d: link 1 is mapped to link d

e: link 2 is mapped to link e

f: link 3 is mapped to link f

Function Code 3 (Status Request)

After ringing and sending the function code, the following protocol (given as occam 2 source) is driven :

```

SEQ
  {{{ receive cluster status
    link.from.NCM ? number.of.clusters
    SEQ cluster = 0 FOR number.of.clusters
      link.from.NCM ? processors.in.cluster.available [cluster]
    }}}
  {{{ receive available special processors (1100, 1200, ..., 1800)
    number.of.special.processors := 0
    link.from.NCM ? special.processor
    WHILE special.processor <> 0
      SEQ
        special.procs.array [number.of.special.processors] := special.processor
        number.of.special.processors := number.of.special.processors + 1
        link.from.NCM ? special.processor
    -- end of special processor list
  }}}
  {{{ receive user allocation of processors
    link.from.NCM ? number.of.users.in.system
    SEQ user = 1 FOR number.of.users.in.system
      link.from.NCM ? number.of.allocated.processors [user]
  }}}
  {{{ receive connections switched at the NCUs for the user
    link.from.NCM ? number.of.NCUs
    SEQ index = 0 FOR number.of.NCUs
      SEQ
        link.from.NCM ? NCU.number
        SEQ channel = 0 FOR 96
          link.from.NCM ? user.connections [channel]
  }}}
  {{{ receive direct processor mapping
    link.from.NCM ? number.of.processors
    SEQ index = 0 FOR number.of.processors
      SEQ
        link.from.NCM ? logical.user.number      -- cf. entry b in user.return.list
        link.from.NCM ? physical.user.number     -- cf. entry k in node.device.map
                                                -- with entry j = PROC.DEV
  }}}

```

OCCAM conventions:

`link.from.NCM ? variable` : Read an integer (32 bits) from the link to the NCM (ringing link) and store it in the integer variable 'variable'. For Helios the link has to be set to dumb mode.

`a <> b` : a not equal b

`[ARRAY.SIZE] INT array` : Integer (32 bit) array of size 'ARRAY.SIZE'.

Function Code 7 (User Data Request)

After ringing and sending the function code, the following protocol (given as occam 2 source) is driven :

```

SEQ
  {{{ receive connections switched at the NCUs for the user
  link.from.NCM ? number.of.NCUs
  SEQ index = 0 FOR number.of.NCUs
    SEQ
      SEQ channel = 0 FOR 96
        link.from.NCM ? user.connections [channel]
      }}}
  {{{ receive direct processor mapping
  link.from.NCM ? number.of.processors
  SEQ index = 0 FOR number.of.processors
    SEQ
      link.from.NCM ? logical.user.number      -- cf. entry b in user.return.list
      link.from.NCM ? physical.user.number     -- cf. entry k in node.device.map
                                                -- with entry j = PROC.DEV
    }}}
  {{{ receive node.device.map
  SEQ
    link.from.NCM ? array.size.1
    link.from.NCM ? array.size.2  --> ( = 2 )
  SEQ node = 0 FOR array.size.1
    SEQ
      link.from.NCM ? node.device.map [node] [0]
      link.from.NCM ? node.device.map [node] [1]
    }}}
  {{{ receive connection.mat
  SEQ
    link.from.NCM ? array.size.1
    link.from.NCM ? array.size.2
    link.from.NCM ? array.size.3  --> ( = 2 )
  SEQ node = 0 FOR array.size.1
    SEQ link = 0 FOR array.size.2
      SEQ
        link.from.NCM ? connection.mat [node] [link] [0]
        link.from.NCM ? connection.mat [node] [link] [1]
      }}}
  {{{ receive user.return.index
  SEQ
    link.from.NCM ? array.size
  SEQ node = 0 FOR array.size
    SEQ
      link.from.NCM ? user.return.index [node]
    }}}
  {{{ receive user.return.list
  SEQ
    link.from.NCM ? array.size.1
    link.from.NCM ? array.size.2  --> ( = 2 )
    SEQ i = 0 FOR array.size.1
      SEQ
        link.from.NCM ? user.return.list [i] [0]
        link.from.NCM ? user.return.list [i] [1]
      }}}
  {{{ receive node.cluster.map
  SEQ
    link.from.NCM ? array.size
  SEQ node = 0 FOR array.size
    SEQ
  
```

```

link.from.NCM ? node.cluster.map [node]
}})

{{{ receive link.mapping
SEQ
link.from.NCM ? array.size.1
link.from.NCM ? array.size.2

SEQ node = 0 FOR array.size.1
SEQ link = 0 FOR array.size.2
SEQ
link.from.NCM ? link.mapping [node] [link]
}})

```

OCCAM conventions:

`link.from.NCM ? variable` : Read an integer (32 bits) from the link to the NCM (ringing link) and store it in the integer variable 'variable'. For Helios the link has to be set to dumb mode.

`a <> b` : a not equal b

`[ARRAY.SIZE] INT array` : Integer (32 bit) array of size 'ARRAY.SIZE'.

Data structures and their meanings for function code 7

```

{{{ node_device_mapping
[MAX.NODES] [2] INT node.device.map :

'node.device.map' stores the physical device for each node.

node.device.map [i] = [j,k] : The node i is mapped to the physical device
                             specified by the index k of the
                             hardware list.
                             The device type is for the list is j.
                             The physical index in HW lists is k.

index i : node (same as 'connecton.mat' index).

entry j : device type
{{{ values
NO.ENTRY   : no map for this node
PROC.DEV   : MC/SC processor
NCU        : NCU device
USER       : User device
ADM.DEV    : NCM administrated device
NOT.ADM.DEV : not NCM adm. dev.
}}})
entry k : device index in hardware list
}})

{{{ connection.mat
[MAX.NODES] [MAX.PROC.LINKS] [2] INT connection.mat :

connection.mat : Matrix for node connections :
                This list stores the wiring of the nodes defined
                from a user request.

connection.mat [i][j] = [k,l] : Node i is connected via link j to node k
                               link l.

index i : a single node
index j : a single link of node i

entry k : - 1 : node i not connected
          0<=k< SIZE connection.mat : node i connected to node k.

```

```

entry l : 0<=l< SIZE connection.mat[i]: node k is connected via
                                         link l.
      else                               : bad entry, if k <> -1
}}}

{{{ user return index
[MAX.NODES] INT user.return.index :

'user.return.index' is the mapping of index in 'connection.mat' (=node)
to the index in 'user.return.list'.

user.return.index [i] = j : The user definition for node i is found in
user.return.list [j].

index i : node (same as 'connecton.mat' index).

entry j : - 1 : no corresponding user data for this node.
          > 0 : index in 'user.return.list' where the user denotation
          for this node can be found.
}}}

{{{ user return list
[MAX.USER.PROCS] [2] INT user.return.list :

'user.return.list' is the list in that the user and interface software
denotation of nodes (processors) is stored.

user.return.list [i] = [a,b] :

index i : number of user processor
          (cf. entry in user.return.index)

entry a : > 0 : internal processor number of user interface software.
          - 1 : entry not valid.
entry b : logical processor number (user's denotation).

}}}

{{{ node cluster mapping
[MAX.NODES] INT node.cluster.map :

'node.cluster.map' gives the mapping of nodes onto the clusters.

node.cluster.map [i] = j : node i is positioned in cluster j.

index i : node (same index as in 'connection.mat').

entry j : cluster number.

}}}

{{{ link.mapping
[MAX.NODES] [MAX.PROC.LINKS] INT link.mapping :

link.mapping :          This list stores the physical links according
                        to the logical links given in 'connection.mat'
                        second array index (2.nd index also of this
                        array).
                        It maps the logical links of each node in
                        'connection.mat' to the physical links of the
                        device represented by this node.

link.mapping [i] [j] = k : The logical link j of the node i
                           is represented by the physical link k of
                           the device 'node.device.map [i]'.

index i : node index (cf.connection.mat)
index j : logical link of node i.
entry k : physical link of the device
          represented by node i.

}}}
}}}

```

3.6.3 Syntax of Network Descriptor

{{{ Syntax of the descriptor :

A network descriptor consists of a number of lines.

Each line has the following protocol syntax:

INT : Length of the descriptor line (in bytes)
If the length is 0, it is the last line of the descriptor.

[] BYTE : Descriptor line:

The descriptor line consists of `keywords` with a defined sequence of parameters behind it.

{{{ further description

Each keyword can only occur once in one line.

A keyword has a significant beginning, the rest of the token is skipped. (eg. You may write `PROCESSOR` or `PROCXXXX` instead of the keyword `PROC`.)

If no keyword occurs in a line, the line is skipped.

An empty line (without at least a CR) finishes the network descriptor. The first line sent is regarded as comment.
}}}

keywords: PROC : keyword for `P r o c e s s o r R e q u e s t`:
{{{ keyword explanation

PROC first.number second.number

The first number is a processor number (eg. from the OCCAM compiler) which has no further significance.

The second number given by the user is the logical processor number of the user.

The meaning of this number is described in chapter 3.3 "Special Requirements for Processor Numbering")

}}}

LINK : keyword for `L i n k C o n n e c t i o n s`:
{{{ keyword explanation

LINK first.number second.number, third.number fourth.number

This keyword expects (at least) four following numbers, they may be separated by a token without numbers in it (e.g. a comma):

first.number: A processor number which has been declared before by the PROC keyword (second.number). This processor gets a link connection to the processor given in the third.number. The link of this processor is given in the second.number.

second.number:link of the processor 'first.number'

third.number: A processor number which has been declared before by the PROC keyword (second.number). This processor gets a link connection to the processor given in the first.number. The link of this processor is given in the fourth.number.

fourth.number:link of the processor 'third.number'

If first.number or third.number has not been declared before, the given connection is skipped.

}}}

}}}

4.1 INSTALLATION ON A PC

The Network Configuration Manager (NCM) and the user interface software for MultiTool and for Helios are provided on two 5¼" MS-DOS HD disks. To install the NCM and the user interface software on harddisk, you need approximately 1.5 MByte disk space.

**Disk 1: Network Configuration Manager
System Software**

Directory structure:

```
a: \mtool\ncm\ncmmtool restricted MultiTool kernel
      \?cxxx      NCM for MultiTool
      \helios\ncm  NCM for Helios
```

where ? = m for MultiCluster 2 and ? = s for SuperCluster, and xxx = 16, 32, 48, 64, 128, ... denotes the number of processors within the machine.

For example, for a MultiCluster 2/32, the NCM for MultiTool is located in the directory »\mtool\ncm\mc32«.

**Disk 2: Network Configuration Manager
User Interfaces**

Directory structure:

```
a: \mtool\ncm\user      Config & Boot (MultiTool)
      \system          occam 2 compiler (MultiTool)*
      \helios\bin       hconfig (Helios)
      \ncm_maps        examples (Helios)
```

The directory structure is a suggestion how to install the NCM at your hard disk. Note that **existing subdirectories and files at the above directories may be deleted during installation**. A NCM update or extension for another machine is possible.

* necessary for MultiCluster 2/32 and MultiCluster 2/64 only

4.1.1 System Software for MultiTool

The system software should be installed for the dispatcher or system administrator only.

To install the NCM system software (on DOS level), change to the harddisk volume where you want to install the NCM, and copy the **directory** »\mtool\ncm« from **disk 1** (drive a:):

```
xcopy a:\mtool\ncm \mtool\ncm /s
```

Now, the system software for T414 and T800 NCUs, the »NCM.control« and a restricted MultiTool kernel are installed at the directory »\mtool\ncm«.

To boot the NCM, first change to the directory »\mtool\ncm\?cxxx«, where ? = m for MultiCluster 2 and ? = s for SuperCluster, and xxx = 16, 32, 48, 64, 128, ... denotes the number of processors within the machine. For example, for a MultiCluster 2/32 the NCM for MultiTool is located in the directory »\mtool\ncm\mc32«.

If you start »mtool«, you are already able to boot the Network Configuration Manager. (If no MultiTool is installed, call »ncm« instead of »mtool«.) Cf. section 2.2.1.1 "Booting" for a detailed description of the boot procedure.

Note : If a full MultiTool is installed on your system, you may delete the complete directory »\mtool\ncm\ncmmtool« and the batch »ncm.bat« at the directory »\mtool\ncm\?cxxx«, as these system files are contained in the standard MultiTool release.

4.1.2 System Software for Helios

The system software should be installed for the dispatcher or system administrator only.

The installation of the NCM system software for Helios requires an installed and ready to run Helios version 1.1 or later.

To install the system software, change to the hard disk volume where you want to install the NCM, start up Helios and copy the directory »helios« of **disk 1** with full directory structure to the helios directory:

```
cd /helios  
cp -r /a/helios/* .
```

By these commands the NCM (files »ncmi?xxx.bin«, »ncm4?xxx.bin« and »ncm8?xxx.bin«, where ? = m for MultiCluster 2 and ? = s for SuperCluster, and xxx = 16, 32, 48, 64, 128, ... denotes the number of processors within the machine) and two booting tools (shell script »setdef« and file »boot_ncm«) are copied to the directory »/helios/ncm«.

Before booting the NCM, set up the default machine:

```
cd ncm

setdef ? xxx
```

where ? and xxx are as above.

These commands copy the NCM files to some default files used by »boot_ncm«.

Now, you may boot the default machine by

```
./boot_ncm -b
```

Cf. section 2.2.2.1 "Booting" for a detailed description of the boot procedure.

Note : The './' is necessary only if there exists an older version of »boot_ncm« at the directory »/helios/bin«. We suggest to delete older versions of »boot_ncm«, otherwise the old »boot_ncm« will be used if you forget the './'.

4.1.3 User Interface Software for MultiTool

To install the NCM user interface software (on DOS level), change to the harddisk volume where you want to install the user interface software, and copy the **directory** »\mtool\ncm« from **disk 2** (drive a:):

```
del a:\mtool\ncm\user\*.* (for updates only)

xcopy a:\mtool\ncm \mtool\ncm /s
```

Now, the user interface software (»Config and Boot«) and several example PROGRAMs are present at the directory »\mtool\ncm\user«. Do not use this directory for your own work, because it may be deleted by installing an NCM update !

If you are acquainted with MultiTool and want to run some of the example PROGRAMs, copy a TOPLEVEL.TKT to the same directory and start MultiTool:

```
copy \mtool\system\toplevel.tkt \mtool\ncm\user
cd \mtool\ncm\user
mtool
```

Note : For a MultiCluster 2/32 or /64, you have to install the extended occam 2 compiler, cf. section 4.1.4 "Extended Occam 2 Compiler", before you run the example PROGRAMs.

The usage of »EXE Config and Boot« is described in detail in section 3.1 "User Interface Software for MultiTool" and subsections.

To make the »EXE Config and Boot« usable from any new MultiTool directory, proceed as follows:

```
cd \mtool\system

start MultiTool by typing »mtool« (and a blank) and
[ENTER FOLD]

get the file handling utilities
(e.g. [AUTOLOAD], [NEXT UTIL] (maybe several times))

create a new fold and name this fold by
»\mtool\ncm\user\conf.cex«

[ATTACH] this fold by the file handling utilities

rename the fold by »CODE EXE Config and Boot«

take the fold ([MOVE LINE]) and place it into the »Toolkit
fold«, subfold »Autoload« or »Tools« ([TOOLKIT],
[ENTER FOLD], [MOVE LINE])

test the success of this procedure by [GET CODE], exit the toolkit fold
and [RUN EXE]; quit the running EXE by typing 'q'

[EXIT FOLD] and [FINISH] MultiTool
```

Now, for every new installation of a MultiTool directory the »EXE Config and Boot« will be present in the toolkit fold.

For old MultiTool directories, proceed the above topics on the respective directories.

4.1.4 Extended Occam 2 Compiler (MC 2/32 and /64)

To install the **extended occam 2 compiler** (for MultiCluster 2/32 and /64 systems only), insert **disk 2** and type the following DOS commands:

```
attrib -r \mtool\system\occam2.cut

copy a:\mtool\system\occam2.cut c:\mtool\system\
      occam2.cut

attrib +r \mtool\system\occam2.cut
```

Note : The original occam 2 compiler will be **overwritten** by this installation.

4.1.5 User Interface Software for Helios

Insert **disk 2** into drive **a:** after you have booted your Helios operating system.

Copy recursively the directory **»/a/helios«** from the disk into your **»/helios«** directory:

```
cd /helios

cp -r /a/helios/* .
```

or copy the disk without example resource maps:

```
cd /helios

cp -r /a/helios/bin/* bin
```

Consider that in the first case existing resource maps on **»\helios\ncm_maps«** may be overwritten!

For an automatic use of **»hconfig«** while booting cf. section 3.2.3 "Automatic Usage of **»hconfig«** on PC Hosted Systems".

4.2 INSTALLATION ON A MULTICLUSTER 1 UNIT

The system and the user interface software are provided on two 3½" disks. The first disk contains the system software and the other the user interfaces. To install the software on the system for MultiTool the following steps are sufficient:

1. Get the disk utilities.
2. Create an empty fold and place the cursor on it.
3. Insert the appropriate disk into the floppy-station.
4. Use the function [STREAM RETRIEVE] to copy the fold structure to the harddisk.

Now the software is installed on the MultiCluster 1 Unit.

The system software should be installed for the dispatcher only.

4.3 INSTALLATION ON A SUN WORKSTATION

The system software and the user interface software for MultiTool and for Helios are delivered on one ¼" streamer tape.

Keep the read/write permissions as they are on the tape.

Note : Please install Helios and/or MultiTool before installing the MultiCluster 2 / SuperCluster software. The installation process of the NCM checks for existing »/mtool« and »/helios« directories.

No superuser permission is required but a **write permission** to »/mtool« and »/helios«.

4.3.1 Main Installation

The main installation process covers the copy of the tape contents to the SUN harddisk. The required disk space is about 6.3 MByte.

Choose a directory for the NCM software. We suggest

```
/home/hostname/ncm
```

After reading the tape, several symbolic links into this directory tree will be created depending upon existing »/mtool« or »/helios« directories.

Following steps are sufficient to install the software on harddisk:

1. Insert streamer tape
2. Create the NCM directory:

```
mkdir /home/hostname/ncm
```

3. Change your working directory thereto:

```
cd /home/hostname/ncm
```

4. Let us assume »/dev/rst0« is your streamer-tape device. Read the streamer tape:

```
tar xvpf /dev/rst0
```

5. For the installation of the NCM software a makefile is provided. Type

```
make install
```

This will install everything. Try the 'make -n install' command for a list of attempted commands.

In order to boot the NCM from MultiTool, the dispatcher or system administrator may change to `»/mtool/ncm/?cxxx«` (where ? = m for MultiCluster 2 and ? = s for SuperCluster, and xxx = 16, 32, 48, 64, 128, ... denotes the number of processors within the machine) and start MultiTool using the command `»ncm«` (cf. section 2.2.1.1 "Booting").

To boot the NCM from Helios, change to `»/helios/ncm«`, call `»setdef ? xxx«` to define the default machine and boot it by `»./boot_ncm -b«` (cf. section 2.2.2.1 "Booting").

The `'./'` is necessary only if there exists an older version of `»boot_ncm«` at the directory `»/helios/bin«`. We suggest to delete older versions of `»boot_ncm«`, otherwise the old `»boot_ncm«` will be used if you forget the `'./'`.

Note : To save disk space, you may delete those directories at `»/mtool/ncm«` and those files at `»/helios/ncm«`, which names contain `'?xxx'` or `'?cxxx'` **not matching** with one of your machines, where ? and xxx are as above.

If you use a MultiCluster 2/32 and/or a SuperCluster 48, you may delete, for example, the directory `»/mtool/ncm/mc16«`, the file `»/helios/ncm/ncm8s64.bin«` etc.

4.3.2 Extended Occam 2 Compiler (MC 2/32 and /64)

If you have installed the NCM for a MultiCluster 2/32 or /64, copy the new occam 2 compiler by

```
make mc32 or make mc64
```

respectively. Note, that this procedure **overwrites** the original occam 2 compiler.

4.3.3 User Interface Software for MultiTool

After the main installation, the user interface software (`»Config and Boot«`) and several example PROGRAMs are present at the directory `»/mtool/ncm/user«`.

If you are acquainted with MultiTool and want to run some of the example PROGRAMs, copy a TOPLEVEL.TKT to the same directory and start MultiTool:

```
copy /mtool/system/toplevel.tkt /mtool/ncm/user
cd /mtool/ncm/user
mtool
```

Note : For a MultiCluster 2/32 or /64, you have to install the extended occam 2 compiler, cf. section 4.3.2 "Extended Occam 2 Compiler", before you run the example PROGRAMs.

The usage of »EXE Config and Boot« is described in detail in section 3.1 "User Interface Software for MultiTool" and subsections.

To make the »EXE Config and Boot« usable from any new MultiTool directory, proceed as follows:

```
cd /mtool/system
```

```
start MultiTool by »mtool« (and a blank) and [ENTER FOLD]
```

```
get the file handling utilities
```

```
(e.g. [AUTOLOAD], [NEXT UTIL] (maybe several times))
```

```
create a new fold and name this fold by  
»/mtool/ncm/user/conf.cex«
```

```
[ATTACH] this fold by the file handling utilities
```

```
rename the fold by »CODE EXE Config and Boot«
```

```
take the fold ([MOVE LINE]) and place it into the »Toolkit  
fold«, subfold »Autoload« or »Tools« ([TOOLKIT],  
[ENTER FOLD], [MOVE LINE])
```

```
test the success of this procedure by [GET CODE], exit the toolkit fold  
and [RUN EXE]; quit the running EXE by typing 'q'
```

```
[EXIT FOLD] and [FINISH] MultiTool
```

Now, for every new installation of a MultiTool directory the »EXE Config and Boot« will be present in the toolkit fold.

For old MultiTool directories proceed the above topics on the respective directories.

4.3.4 User Interface Software for Helios

The Helios user interface software will be installed automatically with the main installation. Further installation cf. chapter 3.2.4 "Automatic Usage of »hconfig« on SUN Hosted Systems".

A Quick Reference

This appendix gives a short reminder for using both the system and the user interface software described in chapters 2 and 3.

Note that loading and running the system software is the task of the dispatcher and done only once after switching on the MultiCluster 2 / SuperCluster. The short instruction is listed in section A.1.

The user on the other hand only has to deal with the user interface software in order to configure and load networks. The necessary steps are shown in section A.2.

A.1 RUNNING AND USING THE SYSTEM SOFTWARE

- 1) Switch on the MultiCluster 2.
- 2) Connect the boot machine with one link of the host transputer (load link) to the ConfigIn link of the MultiCluster 2.
- 3) Start MultiTool or Helios on your boot workstation. The system software is found in the directory »/mtool/ncm« (which should be your current working directory) for MultiTool, and in the directory »/helios/ncm« (files ncm_init.bin, ncmt4.bin and ncmt8.bin) for Helios.
- 4) Load the system software:

MultiTool :

Get the »EXE NCM.control«.

Place the cursor on the fold »NCM-Software ...« and run the previously loaded EXE.

Setup the load link of the host system in the displayed menu: Type <s>* and enter the link number.

Load the programs »NCM.init« and »Tx NCM« (located within the fold »NCM-Software ...«) to the MultiCluster 2 /

* You have to type the key s only rather than typing < s >. The brackets shall indicate that only a single keystroke is necessary to select the option.

SuperCluster by typing <1>. This will transfer all necessary code from the boot machine to the transputer system.

Quit the EXE.

Helios :

```
Type  setdef ? xxx  
      boot_ncm -l y -b
```

where ? = m or ? = s for MultiCluster 2 or SuperCluster respectively,
xxx = 16, 32, 48, 64, 128, ... is the number of processors within the
system and y the number of your load link.

The MultiCluster 2 / SuperCluster is now ready for use. We suggest to disconnect the load link of your boot machine from the ConfigIn. This will release the boot machine for usage as a user workstation.

The request for a transputer network within the MultiCluster 2 / SuperCluster is done by the »EXE Config and Boot network« or by the occam 2 compiler for MultiTool users and by calling »hconfig« for Helios users.

A.2 RUNNING THE USER INTERFACE SOFTWARE

Both Helios and MultiTool users need a specific software each time they want to communicate with the transputer system. Helios uses this software to get the nodes for its topology and MultiTool wants to get access to transputer networks for its PROGRAMs.

A.2.1 Configure a Network and Load a MultiTool User Program

a) MultiCluster 2/16-1 NCU and SuperCluster

Loading an occam PROGRAM on a reconfigurable machine requires establishing on line the specific topology of your program.

For the MultiCluster 2/16-1 NCU and the SuperCluster this is achieved by using the «EXE Config and Boot Network» instead of the compiler function 4 [LOAD NETWORK] :

1. Connect the host transputer with one link to the link interface of the MultiCluster 2 (user entries) or SuperCluster (first user entries).
2. Get the »EXE Config and Boot Network«.
3. Place the cursor on the program fold you want to boot.
4. Run the EXE and follow the instructions:

Type <1> and enter the number of the link which is connected to the Link-Interface of the MultiCluster 2 / SuperCluster.

Choose 'new configuration and boot network' by typing <cr>.

The Network Configuration Manager now starts with partitioning and configuring the network. After successful configuration, the EXE loads the program into the network.

After finishing your work, return your processors to the system by selecting the <r> option. Afterwards, your processors will be available to the applications of other users.

b) MultiCluster 2/32-2 NCU and MultiCluster 2/64-4 NCU

The **MultiCluster 2/32-2** and **/64-4 NCU** require the extended occam 2 compiler utilities to meet the special hardware requirements of these machines (link renumbering).

Loading an occam **PROGRAM** on a reconfigurable machine requires establishing on line the specific topology of your program.

1. Connect the host transputer with one link to the link interface of the MultiCluster 2 (user entries).
2. Get the occam 2 compiler utilities.
3. Place the cursor on the program fold you want to boot.
4. Enter the toolkit fold and the "**Transputer config parameters**" fold to choose the configuration link and compile mode. To start the configuration procedure execute the [CONFIG] function of the compiler utilities.

The Network Configuration Manager now starts with partitioning and configuring the network. After successful configuration, load your program by choosing the right load link and executing the function [LOAD].

After finishing your work, return your MultiCluster 2 processors to the system by starting the »**EXE Config and Boot Network**« and selecting the <r> option. Afterwards, your processors will be available to the applications of other users.

A.2.2 Configure a Network for Booting Helios

The host transputer has to be connected with one link to the Link-Interface of the MultiCluster 2 / SuperCluster (user entries).

If your »loginrc« or »initrc« file already contains a »hconfig« call, you only have to boot Helios in the normal way.

Otherwise, process the following steps:

1. Boot Helios at least on one node.

- 2.1 For **MultiCluster 2/16-1 NCU and SuperCluster**

- 2.1.1 Call

```
hconfig -s -m xxx.map
```

where xxx is your binary resource map file.

- 2.1.2. Call

```
startns xxx.map
```

to boot Helios on the network inside the MultiCluster 2 or SuperCluster.

- 2.2 For **MultiCluster 2/32-2 NCU and MultiCluster 2/64-4 NCU**

- 2.2.1 Call

```
hconfig -m xxx.map -f xxx.rm  
-o xxx_mod.rm
```

where xxx is your original resource map file and xxx_mod is the modified resource map realised on the MultiCluster 2.

- 2.2.2 Call the Resource Map Generator to get a modified .map file for the network server:

```
rmgen -o xxx_mod.map xxx_mod.rm
```

- 2.2.3 Call

```
startns xxx_mod.map
```

to boot Helios on the NCM-modified network on the MultiCluster 2.

Before you shut down the Helios operating system call »hconfig -r« to return the processors to the system. This will make them available to the applications of other users. You can invoke this command automatically using the logout script of Helios.

B Glossary

This glossary will pick up some expressions to serve as a quick reference on the one hand and to explain terms which are not defined in the text on the other hand. References to other items of this glossary are typed in boldface.

application processor

One of the transputers in the processor pool which is not associated with a special task. The number of these transputers available in the system is part of the **MultiCluster 2 / SuperCluster** model notation (e.g. MC 2/32-2 has up to 32 application processors). The contrary is a **special processor** (as the transputer of a GDS, for instance).

boot machine

The **host system** connected to the **ConfigIn link** is called the boot machine. By this connection the **system software** is loaded.

boot_ncm

To load the **NCM system software** within **Helios** the command »boot_ncm« is used. It will load an initial program and the »**NCM**« into the **configuration control system**.

boot workstation

See **boot machine**.

bootstation

See **boot machine**.

Bottom NCU

See **Top NCU**.

C004

The IMS C004 programmable **link switch** is an Inmos VLSI chip which provides a full crossbar switch between 32 link inputs and 32 link outputs. It is used to build up the several link switching sections of an NCU.

channel

A channel in this document shall denote one of the 96 NCU entries. A connected **UniLink** (i.e. a transputer link or an NCU channel) may be switched together with any other NCU entry controlled by the NCU logic.

cluster

A cluster is a collection of **nodes** (transputers or other integrated devices) which build up a unit together with a number of NCUs (typically 1 or 2 NCUs). The number of nodes and NCUs belonging to a cluster depends on the architecture of the transputer system used. The standard **SuperCluster**, for instance, has 4 clusters with 16 **application processors** and 1 NCU each and two clusters with several I/O devices and 1 NCU. A MC 2/64-4 system has 2 clusters with 32 application processors, several I/O devices and 2 NCUs each. Cf. **clustered machine**.

clustered machine

All transputer systems which have at least 2 **clusters** are clustered machines. The following table will show some examples:

system	number of clusters
MC 2/16-1	1
MC 2/32-2	1
MC 2/64-4	2
SuperCluster 64	4

communication link

The data input/ output section of a **UniLink**. Cf. **reset signal**.

Computing Cluster

The **clusters** of a **SuperCluster** built up from 16 **application processors** connected to one NCU. 32 **links** connect the Computing Cluster to outside.

Config and Boot Network

After placing the cursor on a desired **PROGRAM** fold, this **EXE** will communicate with the **NCM**. The **NCM** then tries to configure the **network** and to load the program. A result is sent back to the **EXE**.

config link

Link used by the extended occam 2 compiler utilities running on a **host** machine to configure a network via the [CONFIG] function on a

MultiCluster 2/32 or /64. This link should match with the link of the compiler's [LOAD] function. Don't confuse with the **ConfigIn link**.

ConfigIn link

This is a special **link** entry of a **MultiCluster 2** or **SuperCluster** system. Via this link the **boot machine** loads the **system software** onto the **configuration control system**. In larger systems this entry may also be used to communicate with another machine via their **ConfigOut link**.

ConfigOut link

To build up larger systems **MultiCluster 2** and/or **SuperCluster** systems may be connected. This is done by connecting the **configuration control systems** of these machines via the **ConfigIn** and **ConfigOut links**. Thus, a pipe of all involved NCUs is formed.

configuration control system

The configuration control system (configuration subsystem) of an **electronically reconfigurable machine** is built up by all NCUs of the transputer system. It will distribute and control the resources upon user requests. The software running on it is the **NCM**.

configuration subsystem

See **configuration control system**.

crossbar switch

See **C004**.

descriptor fold

A compiled and extracted **PROGRAM** foldset contains a descriptor fold which assembles the information about the program configuration i.e. the **link** connections and the processors needed for the **network**. This fold is created by the **occam** compiler and sent to the **NCM** during a **network request**.

dispatcher

In this document the person who is authorized to install and run the **system software** or to perform other privileged operations is called the dispatcher or system administrator. He has to have access to the **ConfigIn link**.

This expression should not be confused with **superuser** used in connection with multiuser operating systems.

electronic reconfigurability

See **reconfigurability**.

electronically reconfigurable machine

All machines with a **configuration subsystem** (NCUs + NCM) which may establish various transputer **networks** upon user requests.

ETN

The MTM-ETN is a transputer based Ethernet board designed by PARSYTEC. A software package providing full TCP/IP facilities is available for **Helios**. The ETN can be used as a **special processor**.

Euler Cycle

An Euler Cycle is a special kind of a mathematical graph with the following characteristics:

- every **node** has an even number of edges
- there is a path through the graph which starts and ends at the same node
- on this path every edge is passed exactly one time

In some cases the NCM will generate an Euler Cycle from your desired **network** to find a matching transputer network in systems with **even** and **odd** NCUs

even NCU

To reach **electronic reconfigurability** with a minimum of costs some **MultiCluster 2** models divide the **links** of each transputer (within a **cluster**) among 2 NCUs. 2 links for the even NCU and 2 links for the odd NCU. Thus, a third NCU will be saved.

event

The event is a special transputer signal which allows external hardware to signalize several changes in state. The **T212** of the NCU, for instance, will pull the event signal of the **T414** to notify a user request. The NCM which is implemented in **occam** (running on the T414) will notice the event by channel communication.

EXE

EXE is a key word used with the **MultiTool** system to indicate an executable foldset. Typically this foldset consists of an **occam** source fold, the code fold and a **descriptor fold**. The code runs on a single transputer (i.e. the **host transputer** which runs MultiTool).

Fixed Topology Module

Those applications which don't require **electronic reconfigurability** may use a machine with Fixed Topology Modules (FTM) instead of NCUs. An FTM will make a certain fixed **network** topology available to the user.

FTM

acronym for **Fixed Topology Module**.

electronic reconfigurability

A transputer system which allows the user to switch an arbitrary **network** by software requests is called an electronically reconfigurable system. The reconfigurability is reached by connecting each transputer **link** with a NCU which may switch the links controlled by the NCM. See also **reconfigurability**.

GDS

The Graphic Display Subsystem (GDS) is a transputer driven colour graphics system which may be integrated in a **MultiCluster 2** or **SuperCluster** machine. Note that the transputer of the GDS is treated as a **special processor**. It is part of the **I/O subsystem**.

hconfig

To communicate with the **NCM** of the **configuration control system** from the **Helios** environment, hconfig is used. The main task of hconfig is to ask the NCM for a **network** according to the configuration described in the **resource map**. Usually, hconfig is called within the **loginrc** script.

Helios

Helios is a distributed UNIX-like operating system running on arbitrary transputer **networks**. After establishing a network on an **electronically reconfigurable system**, Helios may be loaded on the entire network. See also **hconfig**, **resource map**, **rmgen**, **startns**.

host

See **hostsystem**

host transputer

Any transputer which is directly connected to the system bus of a **host** may serve as a host transputer. Usually, **MultiTool** or **Helios** is booted on this processor. The **user interface software** is running on this **node**, too.

hostsystem

The host or hostsystem is a computer which may directly access a transputer link (**host transputer**) via its internal bus. It will be used as a file server and for other peripheral tasks (screen, keyboard etc.) by the host transputer. A typical host may be an IBM PC-AT with a BBK-PC and a busless transputer board. To get access to a transputer **network** a **link** of the host transputer has now to be connected with a **user entry**.

I/O subsystem

All peripheral transputer boards which are plugged into an **MultiCluster 2 / SuperCluster** build up the I/O subsystem. In a **reconfigurable system** they may be connected with NCUs. Thus, the features of these **special processors** are completely integrated into the transputer **network** to realize.

Typical boards of this class are **MSC, GDS, TFG** etc. Special plugs at the rear panel of MultiCluster 2 and SuperCluster make adaptation easy.

link

A link is used to connect a transputer with another transputer or an NCU **channel**. This will establish a communication line according to the **UniLink** standard. Note that a (physical) link corresponds to an (logical) edge of a transputer **network**.

link interface

All **links** accessible from outside form the link interface. It consists of the **user entries**, the **ConfigIn** and **ConfigOut links**.

load link

To communicate with the NCM the user has to connect one **link** of the **host transputer** - the load link - with a **user entry**. Via this link the host transputer requests **network** topologies and loads programs onto them.

The **dispatcher** who loads the **system software** from his host transputer has to connect a link - the load link - with the **ConfigIn link** of the **MultiCluster 2 / SuperCluster**.

logical processor number

A logical processor number is a label to specify a transputer **node** in an **occam** program or in a **Helios resource map**. This number should not be confused with the **physical processor number** within the transputer system. However, there are some cases where a specification of a physical node by a logical processor number is necessary. This is done by special numbers. Cf. **special processors**.

loginrc

A **Helios** script file which is automatically started after successful login. It usually contains a **hconfig** and a **startns** call to request a **network** from the NCM and to boot Helios onto it.

main processor

The **T414** of the NCU which runs the NCM is called the main processor. It will get **events** by the **T212** (the **sub processor**) if user requests occur.

master node

The transputer **node** on which the **Helios** system is booted and which is connected with the **host system** is called master node.

matrix crossbar switch

See **crossbar switch**.

MC 2

MC 2 is an acronym for **MultiCluster 2**.

MSC

The transputer based Mass Storage Controller (MSC) is designed to control high speed data acquisition systems and parallel data base structures. It is treated as a **special processor** in the transputer system with all necessary plugs at the rear panel.

MultiCluster 1

The MultiCluster 1 system is designed as a transputer software development station or applications target system. Configuration of processor topologies is reached by jumpers. It may serve as a **host system** or **boot station**.

MultiCluster 2

The MultiCluster 2 (MC 2) system has been developed to support **electronic reconfiguration** under software control for PARSYTEC's large range of busless modules at a highly competitive price/performance ratio. It provides the architectural frame to configure medium size systems with up to 64 **application processors**.

MultiTool

MultiTool is the comprehensive development environment based on the Inmos TDS (Transputer Development System).

communicates with the NCM via a **link** connection between **host transputer** and the **configuration control system**.

network configuration unit

The network configuration unit (NCU) is a hardware component which represents the heart of an **electronically reconfigurable** transputer system. Depending on the number of transputers which have to be switched together to generate a **network**, several NCUs are used to form the **configuration control system**. A single NCU serves as a 96x96 **crossbar switch** with a switch level of 1 from **link** to link.

network request

Using MultiTool a network request is done by running the »**EXE Config and Boot network**« for **MultiCluster 2/16-1 NCU** and **SuperCluster** or the extended occam 2 compiler utilities for **MultiCluster 2/32-2** and **/64-4 NCU**. To get a **network** for running **Helios hconfig** is used.

node

Node is a logical notion for a transputer. It is used in connection with the interpretation of a transputer **network** as a mathematical graph.

odd NCU

See **even NCU**.

occam

Occam is a programming language specially developed for the transputer. It generates highly efficient object code and allows a maximum usage of the transputer's parallel features.

partitioning

To configure a **network** the NCM has to allocate free resources (i.e. free **application** and **special processors**). In a **clustered machine** with several independent users the needed transputers have to be taken from different **clusters**. The distribution of transputers to **clusters** to form the user requested network is called partitioning.

physical processor number

The physical processor number is used to denote a transputer within its real machine environment. These numbers are found in the technical documentation of the particular system. The **user software** allows to ask

the NCM to show the mapping of **logical processor numbers** onto the physical numbers. See also **special processor**.

processing subsystem

The **application processors** in their entirety represent the processing subsystem. The other parts of a transputer system are the **I/O subsystem** and the **configuration subsystem**.

processor

Processor is used as a synonym for transputer in this document.

processor pool

The **application processors** and the **special processors** in their entirety represent the processor pool.

processor slot

The location where a processor board is inserted is called processor slot. The backplane of these slots connects the inserted **application processors** with the NCUs resp. FTMs.

The backplane of the slots for **special processor** boards are equipped with **link** converters. Thus, the user may modify the connection between **configuration subsystem** and **special processors** according to the chosen boards. There are also slots which make special plugs available as for the EDN, GDS, TFG and MSC.

PROGRAM

In the **MultiTool** system **PROGRAM** is a key word which marks a foldset for **occam** programs. It mainly consists of the source fold, the **descriptor fold** and the code fold. Such a program may be loaded onto a transputer **network**.

reconfigurability

There are two kinds of transputer systems: **fixed topology** and **reconfigurable** systems. Only the first one is electronically reconfigurable which means that the user may switch the transputer **links** by software to realize a specific **network**. At any time a new topology may be established to meet other requirements.

Every reconfigurable system consists of a **configuration subsystem**, a **processing subsystem** and an **I/O subsystem**.

reconfigurable multiuser system

A transputer system with a **configuration control system** and a set of **user entries** (typically 8) may be used by several users independently. Each user may request arbitrary **networks** from the NCM according to the system resources.

reset signal

Reset-in/-out section of a **UniLink**. cf. **communication link**.

resource map

The resource map is a **Helios** file which contains a description of a transputer **network** (in terms of **logical processor numbers**) on which Helios will run. The network structure which is sent to the NCM is derived from the binary of this file. Cf. **hconfig**, **rmgen**.

return network

Returning a **network** is the action which is done upon user request to free the bound resources. This will make the transputers available for other users. **Helios** uses the »**hconfig**« program and **MultiTool** the »**EXE Config and Boot Network**«.

ringing

Due to the multiple usage of the **user entry** a special reset procedure is necessary to call the NCM. This action is called ringing. It is performed with each user request. The actual **network** of the user is disconnected (not returned !) and a line to a NCU (**sub processor**) is switched.

rmgen

A **Helios** command which compiles a **resource map** and produces a binary version. This extracted resource map is used by several Helios processes like **startns** and **hconfig**.

SC

In the **MultiTool** system, SC is a key word which marks a foldset mainly consisting of a source fold, a **descriptor** fold and a code fold. The source fold contains the program for a single transputer of a **network**. It may be compiled separately. Several SCs are combined to form a **PROGRAM** which runs on a network of transputers.

SC also is used as an acronym for **SuperCluster**.

single user fixed topology system

Some applications without varying **network** topologies don't need **reconfigurable transputer systems**. They will run on a **fixed topology**. To obtain a fixed topology machine **FTMs** instead of **NCUs** are inserted into a **MultiCluster 2**. Different **FTMs** will realize different standard topologies. Cf. **reconfigurable multiuser system**.

special processor

All transputers associated with particular functions (e.g. **ETN**, **GDS**, **MSC**, **TFG**, etc.) are called special processors. They are inserted into a separate section of the transputer system. By special **logical processor numbers** the user is able to select these transputers directly in the configuration part of his programs or in the **Helios resource map**. This ensures that the **network** will contain the desired processor with the appropriate functionality.

sub processor

The **T212** of the **NCU** which establishes the contact between **NCM** and user is called the sub processor. It will trigger **events** for the **T414** (the **main processor**) if user requests occur.

superuser

The superuser of a multiuser operating system is a person with all privileges concerning access rights etc. The usage of the notion in this document is the same as in **UNIX** and should not be confused with the notion **dispatcher**. The dispatcher needs not to be the superuser but the superuser is of course a dispatcher.

system administrator

See **dispatcher**.

system software

The software which runs the **configuration control system** is called the system software. It is installed once after switching on the transputer system by the **dispatcher**. Usually, the users don't have access to the system software.

T212

The IMS T212 transputer integrates a 16-bit microprocessor with 10 MIPS, four standard transputer communications **links**, 2 KBytes of on-chip RAM, a memory interface and peripheral interfacing on a single chip.

T414

The IMS T414 transputer is a 32-bit microprocessor with 10 MIPS performance, four standard transputer communications **links**, 2 KBytes of on-chip RAM, a memory interface and peripheral interfacing.

T800

Additionally to the **T414** features the IMS T800 transputer has a 64 bit floating point unit and 4 KBytes of on-chip RAM. Since approximately November 1990 Parsytec uses the T800 instead of the **T414** on the NCUs.

TFG

The Transputer Frame Grabber (TFG) is a transputer driven image processing system which may be integrated in a **MultiCluster 2** or **SuperCluster** machine. Note that the transputer of the TFG is treated as a **special processor**. It is part of the **I/O subsystem**.

Top NCU

All NCUs of a transputer system build up the **configuration control system**. They are interconnected with **links** forming a pipe. The NCUs that interconnect **Computing Clusters** are called Top NCUs. Sometimes the Top NCU located in the lower area of a **SuperCluster 64** is called Bottom NCU.

UniLink

All transputer links and NCU **channels** within the systems meet the UniLink interconnection standard which makes them plug-compatible. Each UniLink channel provides data-input, data-output, reset-in and reset-out signals to perform serial transmission of 20 Mbit/sec.

user entry

The user entry of a transputer system is a **link** socket at the back plane which has to be connected with a link of the user's **host transputer**. This together with the **user interface software** will give access to the **configuration control system**.

user interface software

To communicate with the NCM the user interface software has to be used. By specifying the **load link** of the **host transputer** (which runs the user interface software) the software will give access to the functions of the NCM.

C Error Numbers

This appendix handles error numbers coming directly from the NCM system software as return values for a user configuration request.

Error codes of the NCM coming via the system interface link (ConfigIn) are listed in chapter 2.2.3 "ConfigIn Link Protocol".

Error messages of the different interface software packages are described in the specific chapters.

NCM.control	see section 2.2.1.2
boot_ncm	see section 2.2.2.2
Config and Boot	see section 3.1.2
extended occam 2 compiler	see section 3.1.3
hconfig	see section 3.2.8

During configuration attempts by a user interface software (the »EXE Config and Boot Network«, the [CONFIG] function of the extended occam 2 compiler utilities or the Helios program »hconfig«) most errors reported by the NCM are coded with a 4 digit number or in textual form. The error status codes have the following format:

$\pm yyxx$ (error xx in process yy)

The first two digits yy denote the configuration process of the NCM where the error occurred (cf. figure 2.1 "Network Configuration by the NCM").

The last two digits xx give a closer specification of the error occurred.

Values of $yyxx > 0$ report a special configuration status but no error. Values $yyxx < 0$ denote errors during network configuration and a value $yyxx = 0$ signals that the configuration succeeded and no error occurred.

List of Process Numbers

The following list of process numbers specifies the first two digits (yy) of the error code above. They are sorted in ascending order. This order does not correspond with the execution order of the processes during configuration.

- 00 main level of configuration
- 01 read user's configuration request
- 02 release the old user resources in the internal NCM lists
- 03 hardware disconnect of the old user's network
- 04 upgrade of user's graph to a graph with four edges
- 05 search an Euler circle through the upgraded user graph
- 06 mark the edges of the graph with odd and even
- 07 get all free resources of the machine
- 08 compare user's network request with the free resources of the machine
- 09 partition the requested network into the available clusters
- 10 optimise the intercluster connections to use as much odd as even external links
- 11 map logical devices to physical devices
- 12 map of logical links to physical links (one-to-one mapping for MC 2/16-1 and SCs)
- 13 allocate requested devices for the user
- 14 store the computed network data for later user requests
- 15 physical connection of the hardware to install the user's network
- 16 generate feedback for the user
- 17 send feedback to the user
- 18 disconnect communication channel between user and NCM and connect the channel to the configured network
- 19 get available links between two clusters for optimising intercluster connections

List of Error Numbers

The following list of error numbers gives particular information for most of the errors:

- In the top line the error number and the error class which is abbreviated as follows:
 - UE** User Error; this class of errors is induced by some action of the user and may also be recovered by the user.
 - INT** INTerface software error; an interface software error is not in the responsibility of the user. However, a recovery may be possible.
 - HWC** Incorrect HardWare Configuration; there may be a wrong wiring in the system which produces this kind of error.
 - HW** Common HardWare problem.
 - SW** Internal SoftWare errors are usually not recoverable by the user.
 - *** An asterisk placed in front of the error class abbreviation indicates that this error is of special relevance to the user.
- A description of the error.
- The consequences of an error occurred are presented in the section 'result'. In most cases your old network is returned and you don't get a new network.
- Finally some hints for possible recovery strategies are given. This is of course not specified for the internal errors.

Note that in some cases, error recovery needs sophisticated knowledge of the hardware. The technical documentation will support you concerning these errors.

-1 UE

description : On a clustered machine which has to be reconfigured using the Euler method (e.g. MC 2/64-4 NCU). A configuration request for a one-to-one configuration has been started.

result : No configuration can be done.

recovery : Do not use one-to-one configuration (i.e. »EXE Config and Boot« or extended compiler function [CONFIG] resp. with an »NCM desc« or second configuration »hconfig -s«).

Use compiler function [CONFIG] and produce an original »descriptor« before.

Use first configuration »hconfig -f« option.

occam : Set the 'compile' parameter to TRUE within the fold »Transputer config parameters«.

-101 INT

description : Timeout at user channel. Protocol broken.

result : NCM breaks the whole protocol and waits for new configuration.

-102 *UE

description : In the occam program configuration or the Helios resource map more than one processor is defined to connect to the second user entry (e.g. processors 1024 and 1124 are defined).

result : Old network returned, no new configuration done.

recovery : Define only one processor xx24 ($10 \leq xx \leq 18$).

-103 *UE

description : Processor which should be connected to the first user entry (= first processor of the numbering range for system processors defined in the occam configuration or Helios resource map resp.) has no link which is unused in the network.

result : Old network returned, no new configuration done.

recovery : Remove one link connection from the first defined system processor in your occam PROGRAM configuration or in your Helios resource map respectively.

-104 INT

description : Link number given is out of range.

result : Old network returned, no new configuration done.

recovery : Internal error of the interface software.

-105

INT

description : Processor request incomplete. Key word 'PROC' without following processor number.

result : Old network returned, no new configuration done.

recovery : Internal error of the interface software.

-106

INT

description : Link request incomplete. Key word 'LINK' without following link definition.

result : Old network returned, no new configuration done.

recovery : Internal error of the interface software.

-107

HWC

description : NCU channel where the ringing was detected is no user entry defined in the system.

result : If the ringing channel was used by another user, the connected processor of that user got a result.

recovery : Configure the cables of the system in the way which is expected by the system software.

-108

UE

description : Network requirement was inconsistent (e.g. processor defined twice or link used twice).

In a Helios resource map subnets may have been defined with the same processor number in different subnets.

result : Old network returned, no new configuration done.

recovery : Control the required network for double or inconsistent descriptors and correct them.

-109

UE

description : Internal array overflow. Too much processors and devices with different characteristics have been defined.

result : Old network returned, no new configuration done.

recovery : Reduce the number of different characteristics of the processors and devices (e.g. each special processor has its own characteristic).

-110

UE

description : Array overflow. Too much processors and devices have been requested. The machine has less devices than required by the user.

result : cf. -109

recovery : Reduce the number of the processors and devices to the number of processors and devices existent in the machine.

-111

UE

description : Array overflow. Too much (external and internal) nodes have been defined.

result : cf. -109

recovery : Reduce the number of processors of the whole network.

-2xx

description : No errors possible.

-301

HW

description : cf. -306.

-302

HWC, HW

description : The NCU communication pipe of the main processors is broken. Disconnect or 'reset processor' command cannot pass through the pipe (error produced by the pipe communication process).

result : User network cannot be returned, user network partially disconnected; all network resources remain allocated by the user; user is not able to get a new network configuration.

recovery : Try again some times.

The cable between ConfigOut (1st machine) and ConfigIn (second machine) could be removed.

The NCU pipe configuration inside the machine could be broken.
A cable connecting the pipe could be out of order.
You may remove the source of error while the machine is running. The NCM should work again afterwards (since version 2.4).

-303 HWC, HW

description : cf. -302 (direct return of the disconnect command).

result : cf. -302.

recovery : cf. -306.

-304 SW

description : Error in the administration list of tasks to be sent to other main processors or other process levels.

result : cf. -302.

-305 SW

description : Got a respond from wrong NCU main processor.

result : cf. -302.

-306 HW, SW

description : Failed to reset all used processors while disconnecting the (old) network. Possible reasons:

- failed to connect main and subprocessor (HW)
- subprocessor has failed to reset a processor (HW, SW)
- internal list error (SW)

result : cf. -302.

recovery : Shut down the system and retry, if retry fails consult Parsytec or change HW elements.

-307 HW, (SW)

description : Timeout receiving feedback for the disconnect command from at least one NCU (error may have the same reasons as for -301, -302, -303, -304, -305 or others).

recovery : cf. -302 or -306.

-308 HW, (SW)

description : Timeout receiving feedback for command 'reset used processors' while disconnecting the (old) network (error may have the same reasons as for -302, -302, -304, -305, -306 or others).

result : cf. -302.

recovery : cf. -302 or -306.

-401 INT

description : No Euler graph (graph with an even number of edges) found while filling up the graph with 4 edges. The graph description may be faulty.

result : Old network returned, no new configuration done.

recovery : Check the interface software generating a reliable graph description.

-402 *UE

description : The network required is not coherent; the graph consists of at least two separate graphs not connected among one another.

result : cf. -401..

recovery : Change the network description (occam: configuration section of PROGRAM; Helios: resource map) to achieve a coherent network.

-501 SW

description : Internal Euler graph element has 0, 1 or 3 edges (should have 2 or 4 edges !).

result : cf. -401.

recovery : Internal error.

-502 SW, (UE)

description : cf. -402.

-503 SW

description : Error generating Euler cycle; pointer out of range.

result : Old network returned, no new configuration done.

recovery : Internal error.

-504 SW

description : Constructed cycle is no Euler cycle.

result : Old network returned, no new configuration done.

recovery : Internal error.

-505 SW

description : Parameter 'number of nodes' out of range.

result : Old network returned, no new configuration done.

recovery : Internal error.

-506 SW

description : Parameter 'start.node' out of range.

result : Old network returned, no new configuration done.

recovery : Internal error.

-507 SW

description : Internal graph inconsistent (connection matrix).

result : Old network returned, no new configuration done.

recovery : Internal error.

-508 SW

description : Link number in internal graph (connection matrix) out of range.

result : Old network returned, no new configuration done.

recovery : Internal error.

-6xx

description : No error possible.

-701

SW

description : Failed to get list of available resources.

result : Old network returned, no new configuration done.

recovery : Internal error.

-801

*INF

description : Not enough processors available in the system pool.

result : Old network returned, no new configuration done.

recovery : Wait until enough processors are available. This takes place when other users free their resources by returning their networks.

-802

INF, SW

description : Needed NCU channel not available (allocated by the system).

result : Old network returned, no new configuration done.

result : No configuration.

-803

SW

description : Requested user entry not available.

result : Old network returned, no new configuration done.

-804

INF

description : At least one administrated device (e.g. a special processor) is not available.

result : Old network returned, no new configuration done.

recovery : Check if the special processor exists; wait until it is available (another user may have allocated the device).

-805 INF

description : A so called not administrated device is not available.

recovery : Device does not exist, do not use.

-901 UE

description : The network cannot be mapped to the free resources.

result : Old network returned, no new configuration done.

recovery : Try later or reduce the complexity of the graph.

-902 SW

description : Some nodes were not mapped to a cluster during partitioning process.

result : Old network returned, no new configuration done.

recovery : Internal error.

-903 SW

description : Internal protocol in partitioning process bad.

result : Old network returned, no new configuration done.

-904 SW

description : Recursion depress for partitioning too small.

result : Old network returned, no new configuration done.

recovery : Increment and add another row of Clustering Numbers [0,...,0] in "pre.part.map" in hardware depending constants of the hardware lists.

-905 SW

description : Wrong number of nodes received from the lower recursion level.

result : Old network returned, no new configuration done.

recovery : Internal error.

-1001 UE

description : Network not realisable with odd and even NCUs; a link mapping cannot be done because too much links need a fix NCU.

result : Old network returned, no new configuration done.

recovery : The Euler method may fail for devices which do not have the same number of even and odd links. Reduce the usage of such devices or select a different link structure

-1003 UE

description : Too few external links (links between two clusters) left.

result : cf. -1001.

recovery : Wait until another user will return his network.

-1102 SW

description : 'number of nodes' out of range.

result : Old network returned, no new configuration done.

recovery : Internal error.

-1103 SW

description : Node characteristic not valid.

result : Old network returned, no new configuration done.

recovery : Internal error.

-1104 SW

description : 'cluster' not valid.

result : Old network returned, no new configuration done.

recovery : Internal error.

-1152 *UE

description : Processor or device (with special characteristic) could not be mapped to a free resource.

result : Old network returned, no new configuration done.

recovery : Make sure that the processor or device (with that special characteristic) which you requested is existent. If you are sure, wait until that processor or device is made available.

-1201 *UE

description : The link mapping failed. A (even or odd) link needed could not be found.

result : Old network returned, no new configuration done.

recovery : Change link numbers in your configuration or resource map.

-1202 SW

description : 'number of nodes' out of range.

result : Old network returned, no new configuration done.

recovery : Internal error.

-1203 SW

description : Error during request of devices from hardware administration.

result : Old network returned, no new configuration done.

recovery : Internal error.

-1204 SW

description : Wrong tag received from hardware administration.

result : Old network returned, no new configuration done.

recovery : Internal error.

-1205 SW

description : Error during request of links from hardware administration.

result : Old network returned, no new configuration done.

recovery : Internal error.	
-1206	SW
<hr/>	
description : Node number of Euler cycle out of range.	
result : Old network returned, no new configuration done.	
recovery : Internal error.	
-1207..-1219	(UE), SW
<hr/>	
description : Link colours are not consistent.	
result : Old network returned, no new configuration done.	
recovery : Internal error.	
-13xx	SW
<hr/>	
description : xx devices which should be allocated are already allocated.	
result : Old network returned, no new configuration done.	
recovery : Internal error.	
-14xx	
<hr/>	
description : No errors possible.	
-1501	*UE
<hr/>	
description : Some devices or processors were not connectable:	
<ul style="list-style-type: none">- Not enough external links available- The network may not be connectable by odd and even NCU's without using the Euler method (link reconfiguring).	
result : Old network returned. New network partially connected; the network resources become allocated by the user; new random numbers are computed for next network requirement if the random behaviour is omitted.	

recovery : Wait until the (multiuser) status has changed; allocate the non-connectable device using the NCM monitor (cf. section 2.2.1.3). Use link renumbering on MultiCluster 2/32 or /64.

-1502 SW

description : Wrong tag from connection handler.

result : cf. -1501.

recovery : Internal error.

-1503 SW

description : Wrong parameter 'connect.mode'.

result : cf. -1501.

recovery : Internal error.

-1504 SW

description : Internal array overflow.

result : cf. -1501.

recovery : Internal error.

-1505 SW

description : 'connection.mat' inconsistent.

result : cf. -1501.

recovery : Internal error.

-1506 SW

description : A node has not been mapped to a physical device.

result : cf. -1501.

recovery : Internal error.

-1508 SW

description : Link mapping incomplete.

result : cf. -1501.

recovery : Internal error.

-1509

SW

description : Software error in 'connect.handler' (function not known, function mixture, internal list overflow, needed NCU channel allocated, not involved NCU responded: 'connected ok', local task.list inconsistent).

result : cf. -1501.

recovery : Internal error.

-1510

HW

description : Connection at at least one NCU failed (timeout while switching a C004 crossbar switch).

result : cf. -1501.

recovery : Shut down the machine and try again; if a retry fails change hardware section.

-1511

HWC, HW

description : The NCU communication pipe of the main processors is broken; the connect command could not pass through the pipe (error produced by the pipe communication process, cf. -302).

result : cf. -1501.

recovery : cf. -302.

-1512

HWC, HW

description : cf. -1511.

result : cf. -1501.

recovery : cf. -302 or -306.

-1513

*UE

description : A link needed for processor or device connection is allocated (by the system).

result : cf. -1501.

recovery : It is possible that the dispatcher previously has manually connected or allocated this channel. If the connection is necessary, the dispatcher should also allocate the connected device

-15xx with xx > = 20

HWC, HW

description : Timeout receiving feedback for the connect command from at least one NCU. The number of the NCU that did not respond is coded with the last two digits xx of the error number. xx - 20 represents the NCU number.

result : cf. -1501.

recovery : cf. -302.

-1901

SW

description : Error while getting NCU devices from hardware lists; invalid characteristic given.

result : Old network returned, no new configuration done.

recovery : Internal error.

-1902

SW

description : Error while getting connected links between two NCUs.

result : Old network returned, no new configuration done.

recovery : Internal error.

-1903

SW

description : List 'available.cables' is inconsistent.

result : Old network returned, no new configuration done.

recovery : Internal error.

-1910

SW

description : Error while getting cluster numbers for NCUs; invalid NCU characteristic.

result : Old network returned, no new configuration done.

recovery : Internal error.

-1911

SW

description : Number of cluster > 2 and Euler method required; not implemented.

result : Old network returned, no new configuration done.

recovery : Internal error.

D Physical Device Numbers

In this appendix physical numbers of devices, i.e. NCUs, application processors and special processors are listed.

D.1 Physical Numbering of NCUs and Application Processors

The following pictures show the physical location of the application processors and the NCUs within the various MultiCluster 2 and SuperCluster machines:

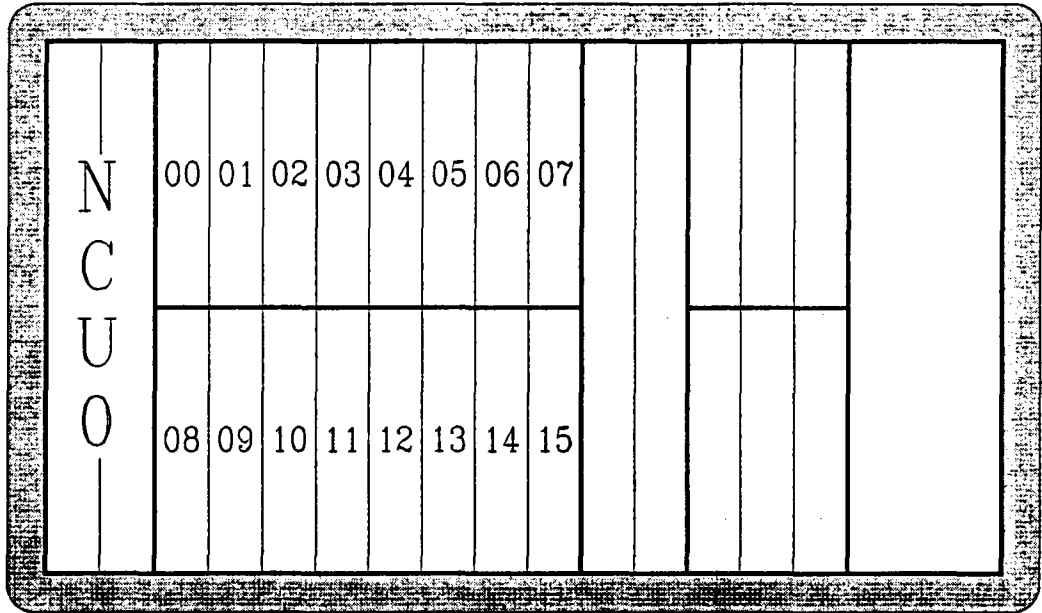


Fig D.1 MultiCluster 2/16-1 NCU front view equipped with TPM-4 boards. Physical processor and NCU numbering by the NCM.

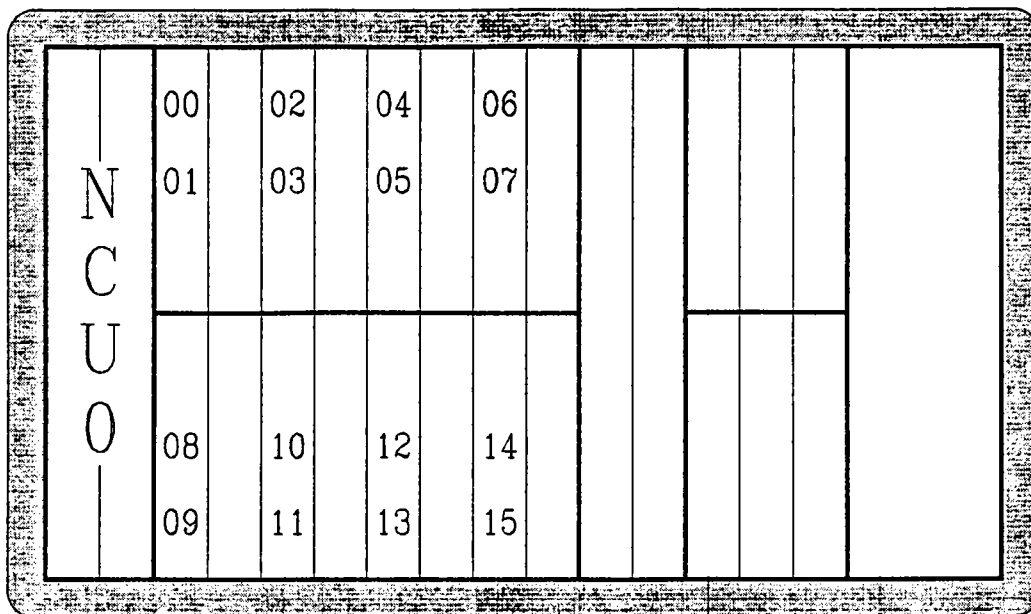


Fig D.2 MultiCluster 2/16-1 NCU front view equipped with MTM-2 boards. Physical processor and NCU numbering by the NCM

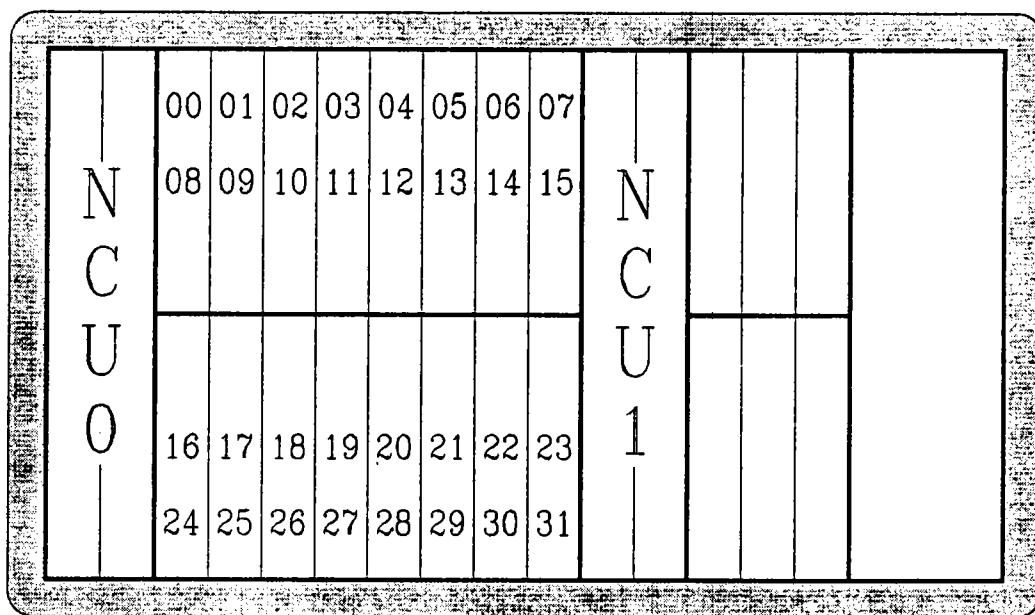


Fig D.3 MultiCluster 2/32-2 NCU front view equipped with MTM-2 boards. Physical processor and NCU numbering by the NCM. All devices are located in Cluster 0.

NCU 0	00	01	02	03	04	05	06	07	NCU 1			
	08	09	10	11	12	13	14	15				
	16	17	18	19	20	21	22	23				
	24	25	26	27	28	29	30	31				

Cluster 0

NCU 2	32	33	34	35	36	37	38	39	NCU 3			
	40	41	42	43	44	45	46	47				
	48	49	50	51	52	53	54	55				
	56	57	58	59	60	61	62	63				

Cluster 1

Fig D.4 MultiCluster 2/64-4 NCU front view equipped with MTM-2 boards. Physical processor and NCU numbering by the NCM.

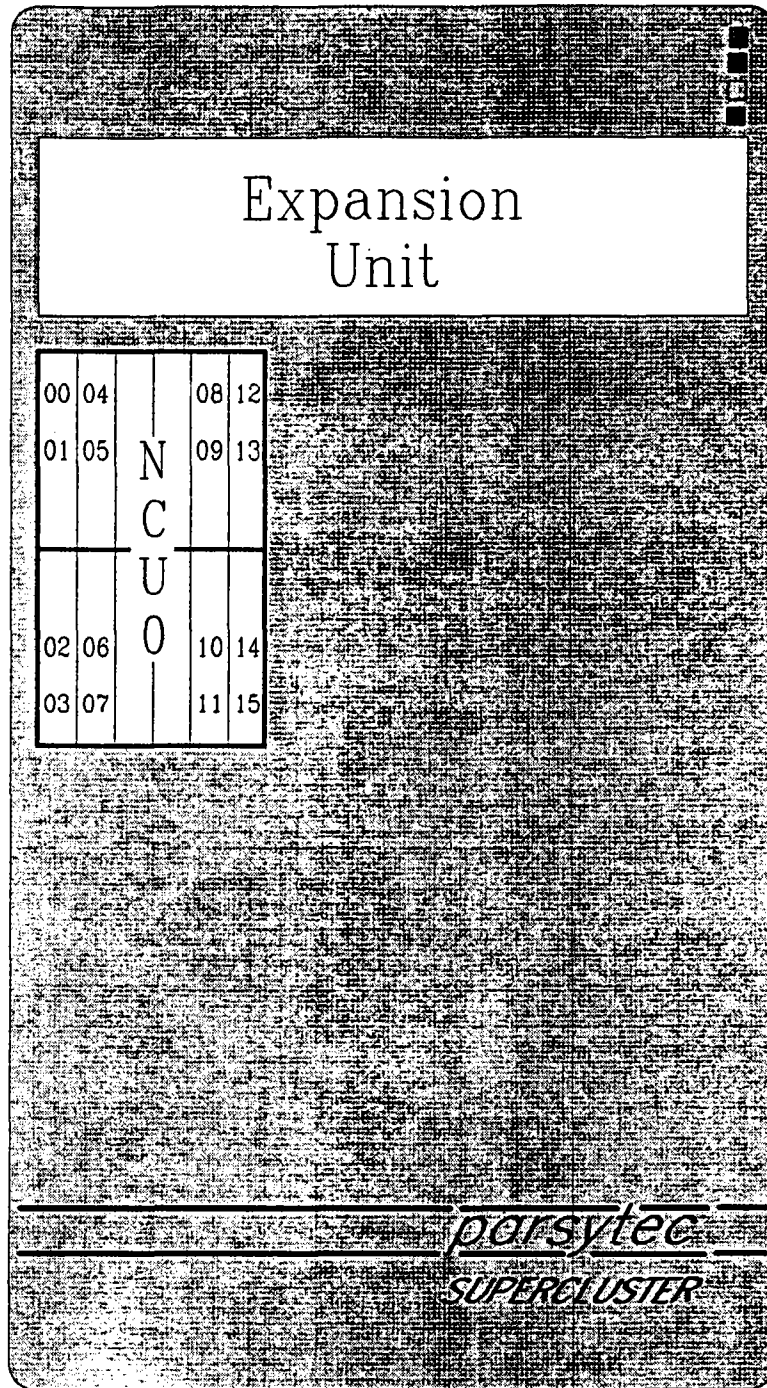


Fig D.5 SuperCluster 16 front view equipped with MTM-2 boards. Physical processor and NCU numbering by the NCM.

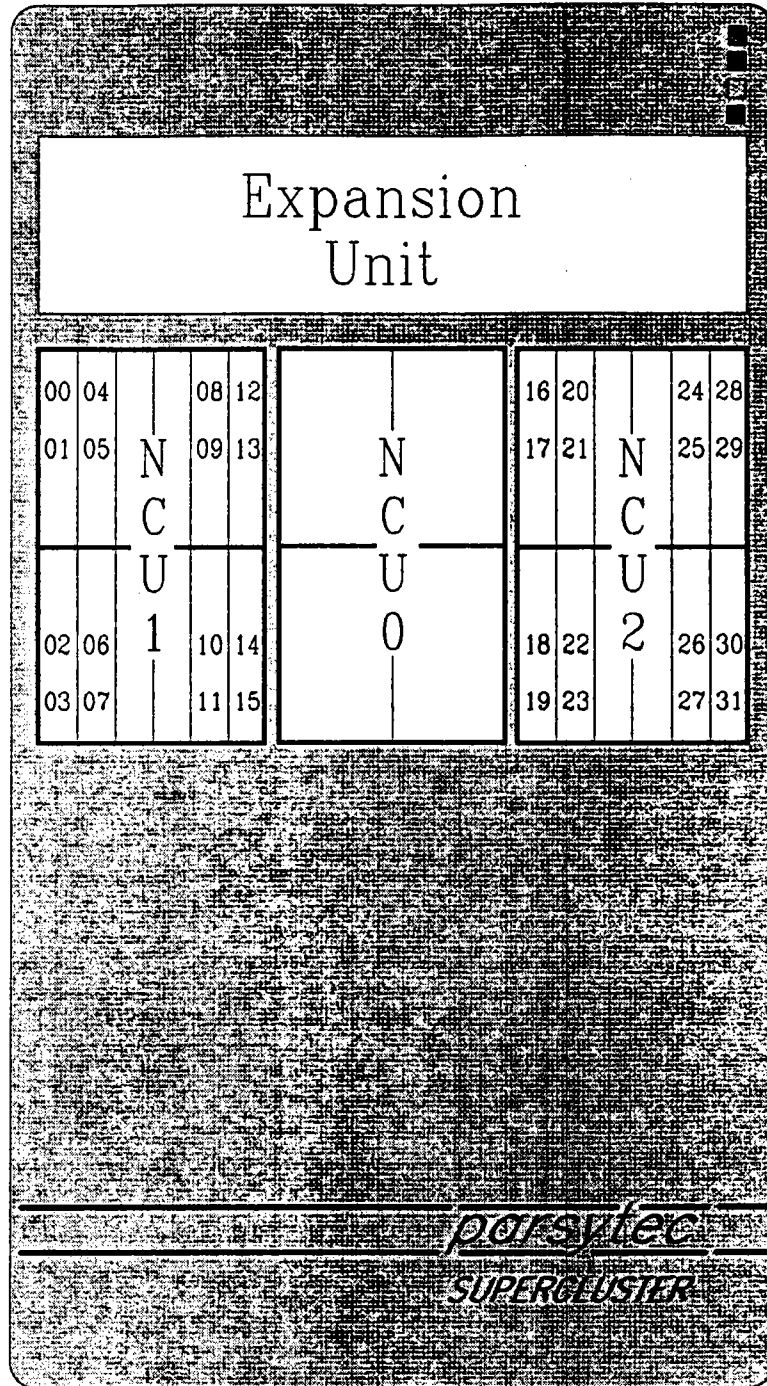


Fig D.6 SuperCluster 32 front view equipped with MTM-2 boards. Physical processor and NCU numbering by the NCM.

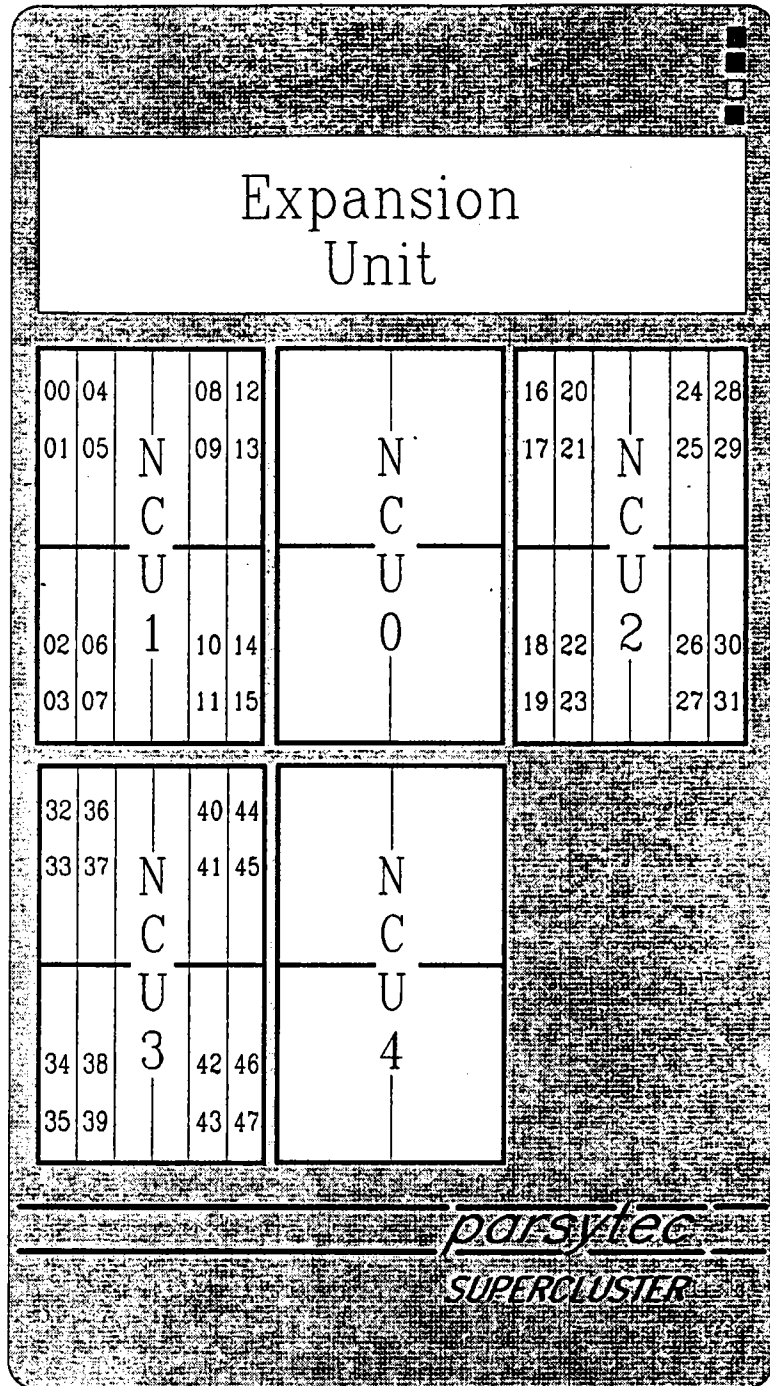


Fig D.7 SuperCluster 48 front view equipped with MTM-2 boards. Physical processor and NCU numbering by the NCM.

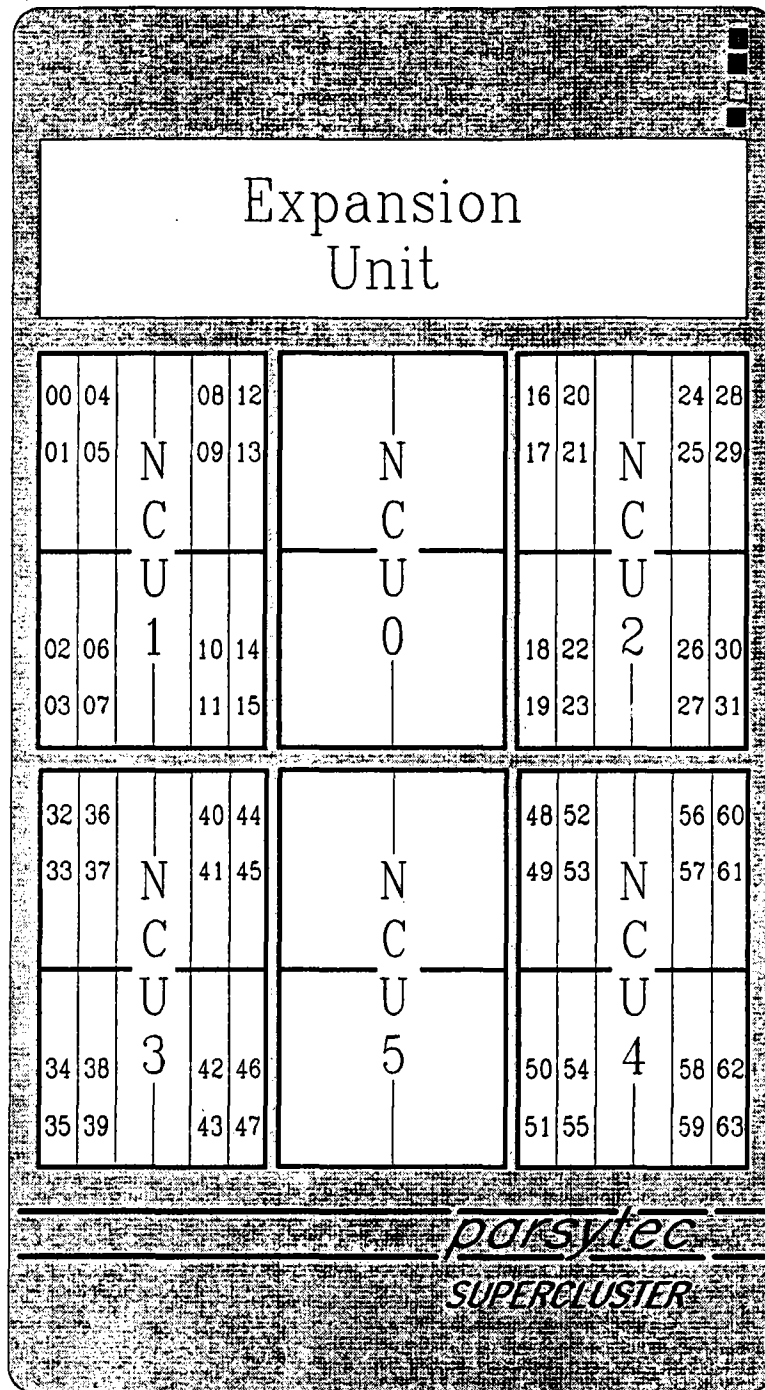


Fig D.8 SuperCluster 64 front view equipped with MTM-2 boards. Physical processor and NCU numbering by the NCM.

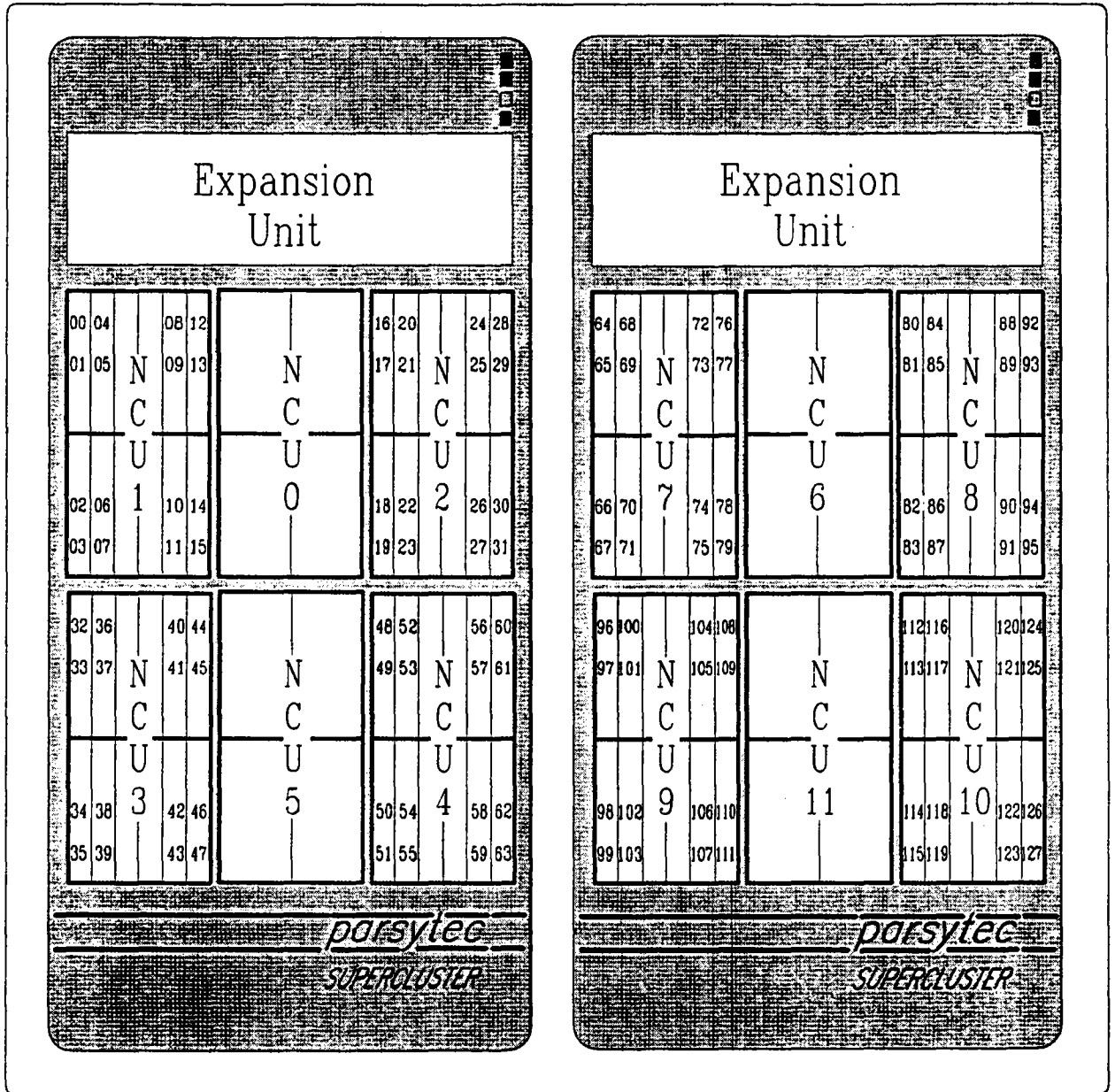


Fig D.9 SuperCluster 128 front view equipped with MTM-2 boards. Physical processor and NCU numbering by the NCM.

D.2 Physical Numbering of Special Processors

Special processors do not have fixed locations within the machines. They are requested via special numbers (1100, 1200, etc.). The physical numbers and the clusters where they are located are given in the following chart:

special processor	physical processor number	cluster number					
		MC 2/16	MC 2/32	MC 2/64	SC 16/32	SC 48/64	SC 128
1100	0	0	0	1	0	0	5
1200	1	0	0	1	0	0	5
1300	2	0	0	1	0	0	5
1400	3	0	0	1	0	0	5
1500	4	0	0	1	-	5	6
1600	5	0	0	1	-	5	6
1700	6	-	0	0	-	5	6
1800	7	-	0	0	-	5	6