

RCSL: 52-AA207
pp: 1 : 16
Author: Fritz
Krogh-Jespersen
Edited: April, 1975

CATS 35
Installation Handbook

Keywords: RC 3500, CATS 35, Communication, Software, Network,
Installation Handbook.

Abstracts: Enclosed in this paper is information about tables, common
variables, and distribution of buffers.

A/S REGNECENTRALEN
3-9, Guldsmædgade
DK-8000 Aarhus C.

CONTENTS

Section

LINKTABLE	1
BUFFER LIST	2
OCCUPIED STORAGE LIST	3
MTC TERMINAL LIST	4
GLOBAL NAMES	5
g-names	5.1
EXAMPLE	6
STANDARD BUFFERS	7
Buffer Size	7.1
Number of Buffers	7.2
CDA	7.2.1
MTC	7.2.2
HICC	7.2.3
LOCC	7.2.4

pp. 1 : 16.

The linktable is used by NCC to route a message to the "next module" on its way to the destinator. The contents of the table decide which destinator can be reached through the actual node, i.e. if the name of the destinator is in the table, the destinator can be reached, otherwise not.

The format of an item in the linktable is:

d1	d2
d3	f
l	
s	

8 bytes = 4 words

where:

d1, d2, d3 are the three characters in the destinator name.

f is a not used byte.

l is "local index", i.e. a value forwarded by NCC to the "next module" together with the message buffer. It is for instance used by HICC to determine which outgoing line to use for the message, and by MTC as reference to MTC TERMINAL LIST.

s is the semaphore address of the "next module", e.g. LOCC.

Two global names are associated to the linktable. g65 is the address of the first item and g66 is address of the last (which must be a zero item). It may be convenient to have some unused items in the table during debugging etc. These should be placed as zero-filled items just before g66.

As NCC, when searching for a match on the destinatorname, uses a linear search, the most frequently used items should be placed from g65 and on.

The BUFFER LIST is used by NCC during initialization to distribute standard NCC-buffers to other modules. NCC uses the subroutine "reserve buffer" to get and initialize a buffer-operation, and then NCC signals the buffer to the owner module.

An item in the BUFFER LIST consists of two words:

s
n

where:

s is the semaphore address of the owner.

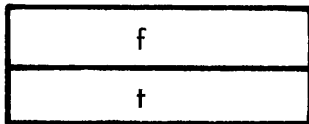
n is the number of buffers wanted by the owner.

The list must terminate with a zero filled word. A global name g67 is the address of the first item.

The occupied storage list is used by the subroutine "reserve space" while it is honouring requests for store from other modules, e.g. "reserve buffer".

In the list are entered 3 items describing occupied areas (except what is occupied by the program).

An item consists of 2 words:



where:

f is the start address of the occupied area.

t is the top address of the occupied area (first
word after area).

For further details see RCSL: 52-AA206 Utilities (Reserve space).

The MTC TERMINAL LIST contains one control operation for each terminal to be automatically enrolled by MTC. The control operation is used by MTC and the STCs to communicate control information such as request for buffer, request for printing, close terminal keyboard etc.

During system reset the operation code must be initialized to MTC basic operation code + 6. An item in the MTC TERMINAL LIST consists of 14 words which at set up time contain:

word

0	0
1	MTC-semaphore address
2	Terminal-semaphore address
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	$1 < 8 + d1$
11	$d2 < 8 + d3$
12	$o1 < 8 + o2$
13	$o3 < 8 + 88$

If terminal semaphore address is different from zero, a "pause" control operation is sent to the terminal (STC) during enrollment (initialization). If it is zero, nothing is sent.

d1, d2, and d3 are the name of the destinator for messages from the terminal, while o1, o2, and o3 are the name of the originator, i.e. the STC routine driving the terminal.

The address of each item must be set up as local index in the LINKTABLE in the entry corresponding to the STC-module (the terminal). One global name (g68) is used as starting address for the MTC TERMINAL LIST.

If the MTC-module is not included in a node, the MTC TERMINAL LIST may be omitted.

To tie the software modules in a node together, it is necessary to use some global names defining hardware configuration, semaphore addresses, standard buffer size etc. A standard in SLAM (the assembler) on RC 3500 is to use f-names for semaphore addresses and g-names for all other purposes. Which of the f- and g-names you use is irrelevant, so when we in this paper use some specific names (g68 etc.), it is only to facilitate the reading of the paper.

g-names.

5.1

g0 up to g31	describe the device numbers for peripherals.
g32 up to g63	describe the corresponding interrupt levels.
g65	address of 1st item in LINKTABLE
g66	address of last item in LINKTABLE
g67	address of BUFFER LIST
g68	address of MTC TERMINAL LIST
g69	address of item in LINKTABLE describing hard-copy routine (used by MTC)
g70	top of free store
g71	address of OCCUPIED STORAGE LIST
g72	number of bytes in a standard buffer (physical size)
g73	NCC operation code
g74	MTC basic operation code (must be different from g73)
g75	number of bytes in an MTC-control operation
g76	number of bytes in a standard buffer preceding SOH

g77	subroutine reserve space
g78	subroutine reserve buffer
g79	subroutine convert binary to octal
g81	subroutine show
g82	subroutine trace
g87	first of DMA-buffer (used by CDA)
g88	top of DMA-buffer (used by CDA)
g91	testward to be used during debugging; reserved values: 1 < 0 trace in Simtimer 1 < 1 show in Simtimer 1 < 2 trace in NCC 1 < 3 show in NCC 1 < 4 trace in MTC 1 < 5 show in MTC
g99	address of a word containing zero (used as bottom of stacks etc.)

The names g77 up to g82 are defined in the e-blocks surrounding the corresponding subroutine.

g87 and g88 need only be defined when CDA and CDI are included in the system. g68, g74, and g75 need only be defined when MTC is included. g69 need not be defined when no hardcopy device exists.

```

D. F127,G127                                ; OUTER SYSTEMBLOCK

;-----
;      INSTALLATION CONSTANTS AND TABLES      ...1...
; 6.3.75      DEMO-SYSTEM
;-----

D. 610                                ; TABLES
    IDR W0 I      81      ; SKIP TABLES DURING RESET

;-----
;      DEVICE NUMBERS
;-----

G0 = 4,      G32 = G0      ; RC810 NR 1
G1 = 5,      G33 = G1      ; RC810 NR 2
G2 = 7,      G34 = G2      ; LAM
G3 = 6,      G35 = G3      ; CDI 100
G4 = 74,     G36 = G4      ; BSC 100-1 READ/WRITE
G5 = 25,     G37 = G5      ; BSC 100-2 READ/WRITE
G6 = 8,      G38 = G6      ; BSC 100-1 KONTROL
G7 = 9,      G39 = G7      ; BSC 100-2 KONTROL

;-----
;      SEMAPHORES
;-----

; F10      CDI 100
; F13      CDA
; F17      NSS
; F19      HTPR
; F25      BSC 100-1
; F26      BSC 100-2
; F32      NCC
; F33      HICC
; F40      PRINT
; F44      MTC
; F45      CLOCK
; F46      SETCLOCK
; F50      RC810 NR 1
; F51      RC810 NR 2
; F60      LAM TERMDRIVER

; TABLES      ...2...

;-----
;      GLOBAL NAMES
;-----

G70 = 8.160000      ; TOP OF FREE STORE
G72 = 8.2600      ; PHYSICAL STANDARD BUFFER SIZE
G73 = 8.3300      ; OPCODE FOR NCC-BUFFERS
G74 = 8.4400      ; MTC BASIC OPCODE
G75 = 28      ; NUMBER OF BYTES IN MTC-CONTROL
G76 = 10      ; REL. ADDR OF SOH IN PHYS. BUFFER
G87 = 8.120000      ; FIRST OF DMA-BUFFER
G88 = 8.122500      ; TOP OF DMA-BUFFER
G91 = 0      ; GLOBAL TESTWORD
G99: 0      ; BOTTOM OF STACKS

;-----
;      OCCUPIED STORAGE LIST
;-----

G71: 8.60000 , 8.100000
      G87 , G88
      G70 , G70

;-----
;      BUFFER LIST
;-----

G67: F13 , 2      ; CDA
      F44 , 1      ; MTC
      F19 , 1      ; HTPR
      F33 , 4      ; HICC
      0

```

; TABICS

...3...

```

;-----
; LINKTABLE
;-----

```

B. A70 , C4

; LINKTABLE

```

; DESTINATOR      INDEX  SEMA      ; NAME  MODULE
G65:
H. 65, 65, 65, 0, W. 0, F13      ; AAA   CDA
G69:
H. 68, 67, 49, 0, W. 0, F40      ; DC1   PRINT
H. 66, 65, 49, 0, W. A1, F44     ; BA1   MTC
H. 66, 66, 49, 0, W. A2, F44     ; BB1   MTC
H. 67, 65, 49, 0, W. 1, F33      ; CA1   HICC LINE 1
H. 67, 66, 49, 0, W. 1, F33      ; CB1   HICC
H. 67, 67, 49, 0, W. 1, F33      ; CC1   HICC
H. 68, 65, 49, 0, W. 2, F33      ; DA1   HICC LINE 2
H. 68, 66, 49, 0, W. 2, F33      ; DB1   HICC
H. 66, 66, 66, 0, W. 0, F19      ; BBB   HTPR
H. 88, 48, 65, 0, W. 1, F13      ; X0A   CDA
H. 66, 66, 67, 0, W. 0, F46      ; B9C   SETCLOCK
H. 88, 48, 71, 0, W. 0, F17      ; X0G   NSS
0, R.8                             ; SPARE ENTRIES
G66:
0, 0, 0, 0                         ; LAST ENTRY

```

```

;-----
; MIC TERMINAL LIST
;-----

```

C0 = G74+6 , C1 = 1<8+65 , C2 = 65<8+65 ; MTC INIT CODE, DESTINATOR NAMES
C3 = 66<8+64 , C4 = 49<8+88 ; ORIGINATOR NAMES AND X

G68:
A1: 0, F44, F50, 0, R.7, C1, C2, C3+1, C4 ; BA1
A2: 0, F44, F51, 0, R.7, C1, C2, C3+2, C4 ; BB1

; TABICS

...4...

```

;-----
; RESET CODE FOR MIC TERMINAL LIST
;-----

```

```

B1:
LDR W1 1 C0 ;RESET CONTROL OP
LDR W2 1 G68
B2:
LDR W3 X2+2
BCW W3 14 F44 , B3
STR W1 X2+12
ADD W2 1 G75
SUB A W0 1 X2+87
B3:
C. ;LINKTABLE
C. ;TABLES

```

A Standard Buffer is a buffer used to pass messages along a link. I.e. it will always pass the NCC, so you might as well call it an NCC-buffer. Buffers used to exchange data between modules need not be standard buffers in regard to size and format, if they are not communicated via NCC (e.g. buffers exchanged between an STC and a terminal driver).

Buffer Size.

7.1

All standard buffers in a CATS 35 system configured by one or more nodes must have the same size, as they must all be able to contain the largest message entered to the system. The buffer size is determined by the following formula:

$$\begin{aligned} \text{buffersize} = & \quad \text{datamax} + \text{messagehead} + \text{messagetail} \\ & + \text{premess} + \text{postmess} \end{aligned}$$

where:

datamax is the largest block of data the system must be able to handle

messagehead is the part of the message containing destinationname etc. (9 bytes)

messagetail is the ETX-character (1 byte)

premess is a work area in front of the SOH-character (at least 6 bytes)

postmess is a work area after the ETX-character (at least 6 bytes).

These two areas are used by HICC, LOCC, and some drivers.

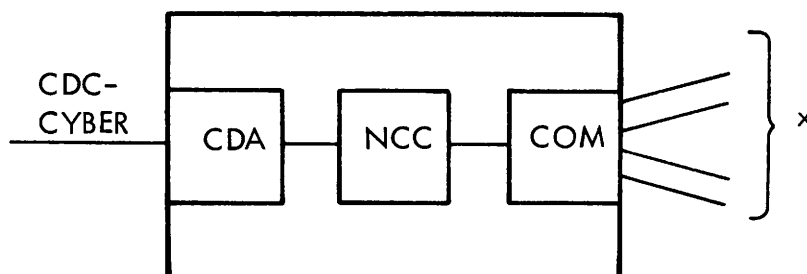
Assume that **datamax** is 1285 bytes. Then $\text{buffersize} = 1285 + 9 + 1 + 6 + 6 = 1307$ bytes = 1308 bytes.

It is often rather difficult to determine how many buffers the different modules should have to ensure a smooth and almost queue-free operation of CATS 35, because the number of buffers is dependent on many factors as line speed, CPU-capacity of the individual nodes, memory size, wanted service level for terminals, the capacity of hosts, and host communication lines etc. Nevertheless, it is in the following tried to give some guidelines for the four main modules CDA, MTC, HICC, and LOCC.

CDA.

7.2.1

Consider a node as shown in fig. 7.1, where x is the number of simultaneously driven lines and COM is a HICC- or LOCC-module.

Fig. 7.1

Assumptions:

- a. All lines run at a line speed of 48K bps.
- b. The line drivers will consume some of the CPU-time, say 10% per line.
- c. A message of 1000 characters occupies CDA for 40 ms. CPU-time.
- d. Transmission of 1000 characters takes 185 ms. real time (incl. acknowledge etc.).
- e. We do not consider incoming message except for the utilization of the line drivers.
- f. We assume that message No. 1 from the CDC CYBER shall be transmitted over line No. 1, message No. 2 over line No. 2, and so on, i.e. message No. $x + 1$ over line No. 1.

The larger x is, the larger the real time used by CDA to prepare a message will be. This is shown in col. 2, table 7.1. The ratio between the number of messages, CDA can prepare while a message is transmitted, and transmission time for a message decreases, when x increases. This is shown in col. 3, table 7.1.

If we want CDA always to have a buffer for preparing a message, the necessary number of buffers is shown in col. 4, table 7.1.

1	2	3	4	5
x	real time per mess.	ratio mess./time	number of buffers	max. delay
1	45	4	2	-
2	50	3.7	3	135
3	57	3.3	4	128
4	66	2.8	4	119
5	80	2.3	4	105
6	100	1.85	3	85

Table 7.1

The assumption f does, of course, not hold, and the maximum delay introduced by an uneven distribution is (transmission time for a message - preparation time for a message). These figures are listed in col. 5, table 7.1.

The assumption d is not true, when the line is busy with incoming messages, or the next node is busy. These situations will increase the maximum delay, or increase the need for buffers. On the other hand, incoming messages will occupy CDA for some time (violation of assumption e), thereby reducing the need for buffers.

Conclusion: A system with a number of buffers equal to 4 (x larger than 2) will give a reasonable efficient system introducing only momentary delays. The system performance cannot be increased drastically without an unlimited number of buffers.

The MTC has two types of buffers, print buffers and communications buffers.

The number of print buffers is determined by the number of terminals which shall be able to continue processing immediately after entering a printrequest to MTC. As printing is a rather seldom event (say one printing per terminal each 10 minutes), one buffer should be adequate for up to 8 terminals and two buffers for a larger number of terminals on the same MTC. We assume a printing to last for 20 seconds. This should give practically no queue.

The number of communication buffers depends on the number of lines on the node, over which messages from MTC shall be transmitted. A convenient number of buffers is one plus one per line.

The HICC-module must have at least one buffer per simultaneously driven line. When one of these buffers is filled, HICC must have a buffer to replace it, if polling must continue on that line. Whether HICC should have still more buffers to be used as replacement for full buffers, depends on how soon HICC will get its buffers back. Normally, the return will take place within say half a second, so an eventual delay will be rather small. On the other hand you may say that if the return of buffers lasts a long time, there is no reason to take new messages into the node, because obviously the node cannot get rid of them. (Remember that a buffer is always returned within finite time).

A reasonable amount of buffers could be:

number of lines	number of buffers
1	2
2	4
3	5
4	6
5	7
6	8

The LOCC module must have at least two buffers to run at all. The first one of these is reserved at reset time by the LOCC itself, so here we will only consider the buffers reserved via BUFFERLIST. Normally, the LOCC-buffers are emptied very fast, either by a HICC in the same node or by a secondary centre, so an adequate number of buffers in BUFFERLIST even for a high traffic rate is 2.