Field Engineering Manual

System 8870

Models 2, 4 and 6

Description of the program interfaces

5.79

1st Edition

Order Number: S 0402 101 05 79

NIXDORF
COMPUTER

Table of contents

1 Module Supply Block

Each micromodule is identified by a module supply block located at the start of the module.

The information on the module supply block is required in the initialisation phase to create the long-time areas and the table (program unit to long-time area, long-time area to short time area, etc.). This information is also required when calling the program unit in during a program run (does dynamic short time area have to be requested? must the program unit be called at its initialisation address? etc.).

Word	Rel. addr.	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit
1	0				module number											call allowed	time characteristics			
2	2			length of module in number of memory words																
3	4			submodule number								quantity of submodules								
4	6	initialisation start address, relative to the start of the module supply block																		
5	8	length of long-time area ⩾ 6						length of the statically assigned short time area												
6	10	logical DMA no., otherwise zero										quantity of queues minus 1				length class, dynamic short time area				

* This bit has a special meaning, and is evaluated by MA 8870.
MA 8870 = module, which interprets the (Assembler 8870) macro-instructions.

Example:

Module Supply Block for NP 0817.01

```
0  0  6  4  0  MVBHD   6.4.4                MODULE NO. 100
0  0  2  5  2          HDF IN-MVBHD+2       MODULE LENGTH
0  0  1  0  1          1....1                    1
0  0  1  8 14          INHID-MVBHD          START ADR INIT.
0  8  0  0  0          8...                 LENGTH OF SHORT TIME
                                             AREA
0  0  0  0  2          2...                 SHORT TIME CLASS 2
```

Module Supply Block for ZD 0811.01, 0816.01, to DMA 1804

```
0  0  6  8  4  MVBSZ   6.8.4                MODULE NO./SIMULTAN.
0  0  3  3 10          SZF IN-MVBSZ+2       MODULE LENGTH
0  0  1  0  1          1....1               SUBMODULES
0  0  2 14  6          INISZ-MVBSZ          INI. CALL-IN
0 12 00  0  0          12...                LENGTH OF SHORT TIME
                                             AREA
0  0  1  0  3          1...3                DMA BIT, SHORT TIME
                                             CLASS 3
```

- Word 1

Bit	
1	not used
2	Bit 2 = 0 } the module has central time characteristics Bit 3 = 0 }
3	Bit 2 = 0 } the module has simultaneous time characteristics Bit 3 = 1 }
4	= 1, the module may be called by means of the call instructions (change from UP to OP Program).
5	
6	The module number is given for purposes of identification at the time of initialisation and during normal program running.
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	The peripheral element is entitled to request a job field and to load this from the UP level.
17	not used
18	

- Word 2

  Bits 1 to 16 give the length of the module in the number of memory words. This information is necessary in order to find the start address of the next module.

- Word 3

Bit	
1	Quantity of submodules If a module extends over more than one memory block it must be broken down into submodules, both in the case of 2 k directly addressable memory areas and in the case of block structures where external means of storage are involved. The quantity of submodules and the submodule numbers must be known in order to tell, before the start of the module, whether it is complete (quantity of submodules) and, where applicable, whether the submodules are in corect sequence in the memory (submodule numbers). Furthermore, a module supply block is to be created for submodules, although this only consists of words 1 to 3.
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	Submodule number
13	
14	
15	
16	
17	not used
18	

Example: If word 3 reads 0.0.1.0.1, there is only one, undivided module.
If word 3 reads 0.0.1.0.2, the module consists of two submodules, of which this is the first.
In the case of the second submodule, word 3 would read 0.0.2.0.2.

- Word 4 Initialisation Call Address

  The initialisation address is called when initialising on the occasion of time or parity interruptions or when starting up again after power failure. The address is given relative to the start of the module supply block.

- Word 5

Bit	
1	Length of the statically assigned short time area. Every program unit allocated to this module is assigned a short time area on initialisation, the length of which is given in bits 1 to 12. If bits 1 to 12 = 0, the module's terminals are working with a dynamic short time area. In that case, the short time area class information is given in word 6 of the module supply block.
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	Length of the long-time area. Every program unit is assigned a long-time area at the time of initialisation. This remains permanently available to it.
14	
15	
16	
17	
18	

- Word 6

Bit	
1	On initialisation, additional short time areas are created. Their lengths are specified in the system parameters. 15 different length specifications (classes) are possible. Should it require it - and if entitled - any program unit of module can request one of these areas through the job connector. Dynamic short time areas exclude static short time areas.
2	
3	
4	
5	This gives the number of queues, minus 1, of each program unit of this module which are to be engaged in their temporary work areas. The queues record the start addresses of the job fields which the program unit has to process.
6	
7	
8	
9	If the module is working with DMA, the allocation of the module to the LDMA is given a DMA number, since there may be more than one DMA in the system. The first such allocation is given no. 1, the second no. 2, etc. This information can be replaced for individual program units of this module by information in the equipment parameters, if bit 7 is set in the index of these parameter. In this case, the logical DMA number appears in the first word after the sequence parameters in the set if equipment parameters. The term DMA refers to DMA 1804 or I/O 1802.
10	
11	
12	
13	
14	
15	
16	
17	
18	

For notes

2 Long-Time Area

The long-time area is assigned to the program unit on initialisation and remains available to it from then onwards.

The long-time areas are called by the "Program unit to Long-Time Area" table (YTNLTB, address 2.4.8 in the case of 8870/4 and 2.1.0 in the case of 8870/6).

Word	Rel. addr.	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit
XLW1	0			error thr. I/O	error inter.	parity inter.	quantity of parameters form XLW 7						inhib bit						PU active	
XLW2	2	module supply block start address																		
XLW3	4	short time area start address																		
XLW4	6	in the case of simultaneous program units, logical DMA no., otherwise zero									quantity of queues minus 1			length class of the dynamic short time area						identical to word 2 in the UP
XLW5	8	loading number for the UP, = 0 in the case of central program units																		
XLW6	10			own program unit number							in the case of central prog. units, priority in bits 1-8, otherwise 1-4					I/O INT 2	I/O INT 1	bell INT 2	bell INT 1	
XLW7	12	bell no. (instruction counter 1)								0	I/O interrupt no. (instruction counter 1)								0	given only if available
XLW8	14	bell no. (instruction counter 2)								0	I/O interrupt no. (instruction counter 2)								0	
XLW9	16	further parameters as given in Word 1, bits 8 to 13																		
XLW10	18																			
	.																			
	.																			
XLWn	n																			

- Word 1

Bit	
1	"1", the program unit is being processed (is active).
2	not used
3	
4	
5	
6	
7	"1", the program unit is inhibited for either interruption process (I/O or bell).
8	Quantity of parameters to be taken over into the short time area. The parameters start from word 7.
9	
10	
11	
12	
13	
14	"1", during job processing, there has been a parity interruption.
15	"1", during job processing, there has been an error interruption (time error).
16	"1", a parity or time error has occurred during data input or output.
17	not used
18	

- Word 2

  This contains the start address of the module supply block, and is required for the program unit first call-in.

- Word 3

  This contains the start address of the associated short time area, but only when the program unit is active. The start address is loaded by the job scheduler at the time of job scheduling.

- Word 4

Bit	
1	This gives the length code of the dynamic short time area which is to be made available to the program unit, if the program unit has been assigned one. Allocation emanates from word 6 (bits 1 to 4) of the module supply block. If bits 1 to 4 ≠ 0, the program unit has been allocated a dynamic short time area.
2	
3	
4	
5	Quantity of queues which are to be allocated to the program unit, minus 1. The queues are created in the short time area. The start addressed of the job fields to be processed are filed in the queues.
6	
7	
8	
9	This gives the logical DMA number in the case of simultaneously operating program units (distinction between the individual program units). It reads 0 in the case of non-simultaneous program unit. This reads 0 in the case of simultaneous program unit working without DMA.
10	
11	
12	
13	
14	
15	
16	
17	not used
18	

- Word 5

  This contains the time load when the program unit is operating simultaneously.

18 13	12 7	6 1	Bit
additional load numbers	additional load numbers	load numbers for program unit central call	

The job scheduler takes over time management.

Every simultaneous program unit in MS 8870 (MS = modular system) is allocated at least one load number "l", with 0 = i = 18. This number is established when generating the System according ot the configuration, the uninterruptable processing times, the program unit's DMA assignment and clock times, and the memories used by the program.

Special points to be considered are:

a) i = 0: the program unit should always be able to operate, that is to say that simultaneous time is henceforth reserved for it in the system.

b) Simultaneous program units which are working with DMA require the following information:

- logical DMA no. Z
- a maximum of two load numbers for operation with DMA assignment
- one load number for operation without DMA assignment.

Example: 3 magnetic tape transports, 1 card reader operated with DMA. The logical DMA no. of all three is Z = 1, and the load numbers are:

Device	load no.	with DMA	without DMA
Tape 1	2	3	1
Tape 2	2	3	4
Tape 3	2	3	5
Card reader	7	-	6

Load numbers without DMA operation must be different.

Load numbers with DMA assignment are the same for the same load.

c) Simultaneous program units with differing time requirements may specify up to three (different) load numbers.

Program units to which neither a), b) nor c) apply are to be clearly identified by a load number.

In the case of program units with DMA operation, the DMA numbers and the load numbers are codified in the installation parameters. Information on this comes from the index (bits 7 and 8).

- Word 6

Bit	
1	"1", the program unit is working with bell interrupt 1. the bell interrupt no. is in word 7.
2	"1", the program unit is working with bell interrupt 2. The bell interrupt no. is in word 8.
3	"1", the program unit is working with I/O interrupt 1. Valid as an alternative to bit 1.
4	"1", the program unit is working with I/O interrupt 2. Valid as an alternative to bit 2.
5	In the case of centrally operating program units, bits 1 to 8 give the priority with which the program unit is entered in the job scheduler's output queue. In the case of simultaneously operating program units, priority of processing is established by means of the bell and I/O interrupt numbers.
6	
7	
8	
9	This contains information concerning the individual number.
10	
11	
12	
13	
14	
15	
16	
17	free
18	

- Word 7

Bit	
1	free
2	I/O interrupt no. 1. This I/O interrupt no. is allocated to instruction counter 1 in short time word 5.
3	
4	
5	
6	
7	
8	
9	
10	free
11	
12	Bell interrupt no. 1. This bell interrupt no. is allocated to instruction counter 1 in short time word 5. It is possible to work simultaneously with I/O interrupt no. 1 and bell interrupt no. 1.
13	
14	
15	
16	
17	
18	

- Word 8

Bit	
1	free
2	
3	I/O Interrupt no. 2.
4	This number is allocated to instruction counter 2 in short time word 6.
5	
6	
7	
8	
9	
10	free
11	
12	Bell interrupt no. 2
13	This bell number is allocated to instruction counter no. 2 in short time word 6.
14	It is possible to work simultaneously with I/O interrupt no. 2 and bell interrupt no. 2.
15	
16	
17	
18	

From word 9 onwards there are sequence parameters which are taken over from the equipment parameters.

Example: Board select address, output bit configuration, flag data, etc.

For notes

3 Short Time Area

The short time area contains the program unit's present working data when it is active.

It can be static or dynamic. The static short time area is created during the initialisation phase before the relevent program unit's long-time area and remains available to it during the active time.

The short time area start address is to be found in long-time word 3.

The dynamic short time area is requested only when the program unit's call-in phase is reached, and is released again by the program unit once the job has been completed. The dynamic short time areas are managed by the system. They are located via the short time class indicator.

Class Indicator	Address 8870/4	8870/6
YKZK 1	2.5.E	2.2.A
YKZK 2	2.6.0	2.2.C
YKZK 3	2.6.2	2.2.E
YKZK 4	2.6.4	2.3.0
YKZK 5	2.6.6	2.3.2
YKZK 6	2.6.8	2.3.4
YKZK 7	2.6.A	2.3.6
YKZK 8	2.6.C	2.3.8
YKZK 9	2.6.E	2.3.A
YKZK 10	2.7.0	2.3.C
YKZK 11	2.7.2	2.3.E
YKZK 12	2.7.4	2.4.0
YKZK 13	2.7.6	2.4.2
YKZK 14	2.7.8	2.4.4
YKZK 15	2.7.A	2.4.6

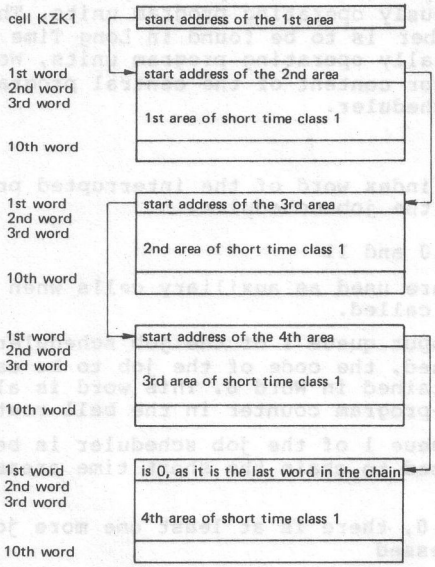
Short Time Area

Word	Rel. addr.	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit
1	0	chaining address of short time areas, or zero																		
2	2	start address of the job field at present in use																		
3	4	auxiliary cell																		
4	6	start address of the long-time area																		
5	8	instruction counter 1																		
6	10	instruction counter 2																		
7	12	index word																		
8	14	auxiliary cell 1																		
9	16	auxiliary cell 2																		
10	18	auxiliary cell 3/chaining of short time area in scheduler, input queue 1																		
11	20	auxiliary cell 4																		
12	22	start address of queue 1																		
13	24	end address of queue 1																		
14	26	start address of queue 2																		
15	28	end address of queue 2																		
.	.	start address of queue n																		
.	.	end address of queue n ($n \geqslant 16$)																		
.	.	During job connection, the quantity of words given in long-time word 1, bits 8-13, is transferred from the long-time area to the (dynamic) short time area by the job scheduler.																		
KW n	n	further auxiliary cells																		

Chaining of areas of the same short time class

The areas are chained via the first word of each individual area.

Example: Class 1 is to be created four times, and class 1 is to be ten words long.



- Word 1

  Refers to the start address of the next free short time area in a chain, and contains zero when the short time area is the last in the chain. It also contains a zero when the short time area is active.

- Word 2

  Start address of the current job field.

- Word 3

  Used as an auxiliary cell for example when further job fields are being scheduled.

- Word 4

  Contains the start address of the long-time area.

- Word 5 (Instruction Counter 1)

  Contains the present interrupt entry address 1 in the case of simultaneously operating program units. The corresponding interrupt number is to be found in Long Time Word 7.

- Word 6 (Instruction Counter 2)

  Contains the present interrupt entry address 2 in the case of simultaneously operating program units. The corresponding interrupt number is to be found in Long Time Word 8. In the case of centrally operating program units, Word 6 contains the accumulator content of the central program interrupted by the job scheduler.

- Word 7

  Contains the index word of the interrupted program, e.g. when calling the job scheduler.

- Words 8, 9, 10 and 11

  These words are used as auxiliary cells when certain sub-programs are called.

  e.g. When input queue 1 of the job scheduler is being processed, the code of the job to be carried out is contained in Word 8. This word is also used as sub-program counter in the bell routine.

  When input queue 1 of the job scheduler is being processed, Word 10 is used to chain the short time areas to one another.

  If Word 10 $\neq$ 0, there is at least one more job in queue 1 to be processed

  If Word 10 = 0, there are no further jobs in the queue.

- Word 12

  From Word 10 onwards the queues are created for the program unit. The number of queues is stipulated in Word 6 of the module supply block.

  In the case of dynamic areas, further data may be taken over from the long-time area (from Word 7) after the queues data. This does not apply to static short time areas.

  After that, further words follow as program unit work cells, e.g. register label, macro interpreter, disk drive work cells, etc.

4 Job Fields

The purpose of job fields is to connect jobs from one program unit to another. They are generated by intelligent program units (macro interpreters) when they want to send a job to an unintelligent program unit.

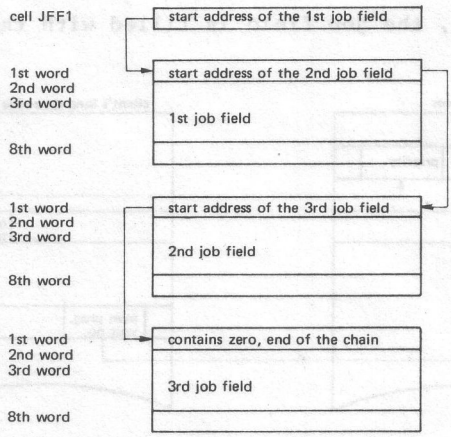
Job fields are memory areas which are dynamically managed by the system and allotted when a request comes from the appropriate program unit. The job fields are located via cell YJFF1.

YJFF1, address 8870/4 2.5.C, 8870/6 2.2.8.

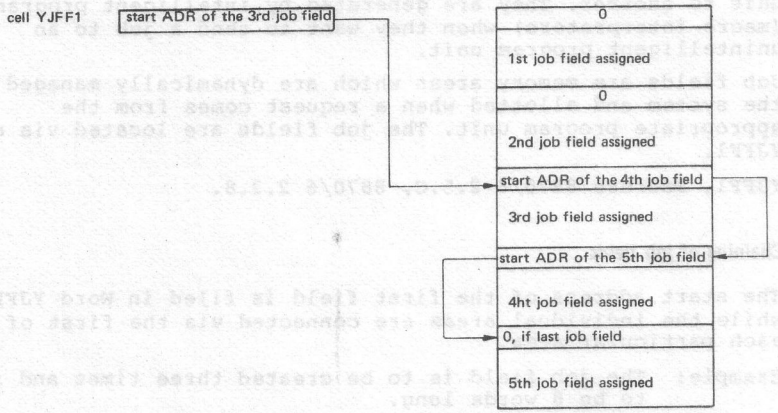
Chaining of job fields

The start address of the first field is filed in Word YJFF1 while the individual areas are connected via the first of each particular area.

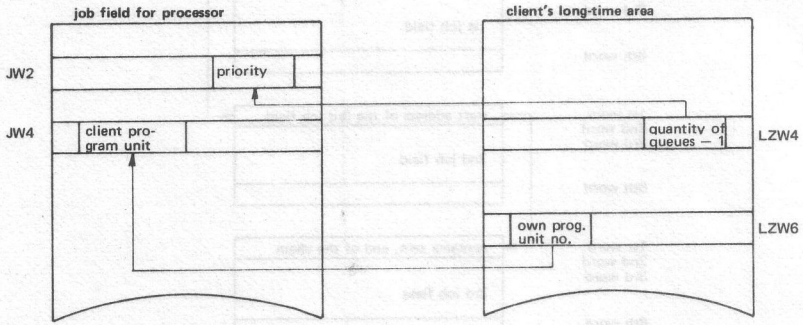
Example: The job field is to be created three times and is to be 8 words long.



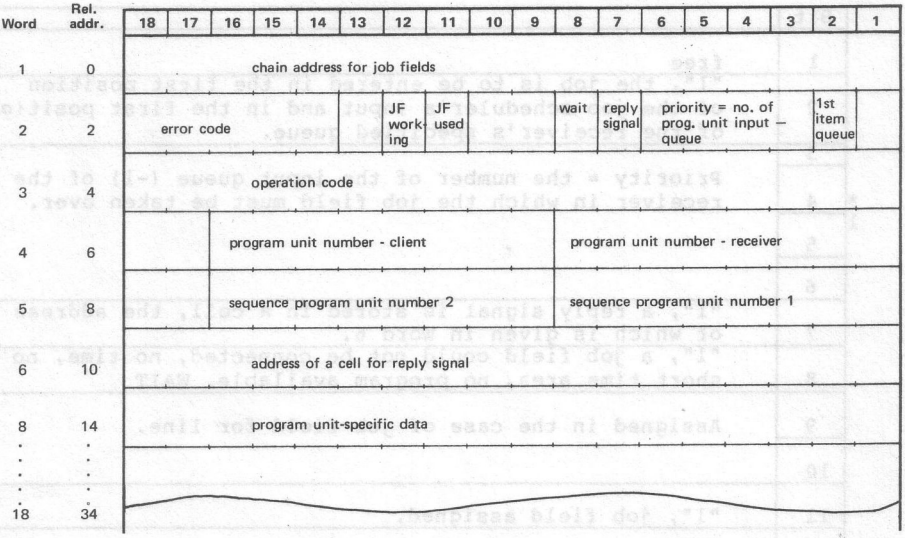
Chaining of job fields according to job allotment



After allotment, the job field is filled with the work data by the client.



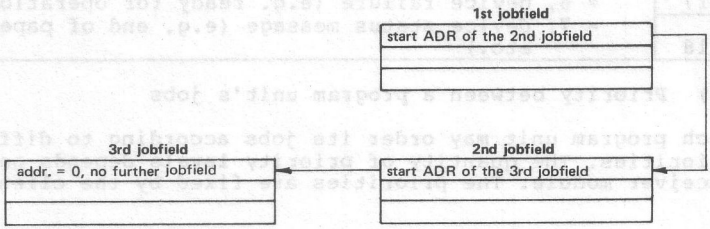
Job Field



A job field normally 19 words long, expect for the controlling program unit's job fields, which are 20 words long.

- Word 1

  Refers to the start address of the next free job field in a chain and contains zero if the current job field is the last in the chain. It also contains a xx zero when the job field is being processed.



- Word 2 (Job Field Status)

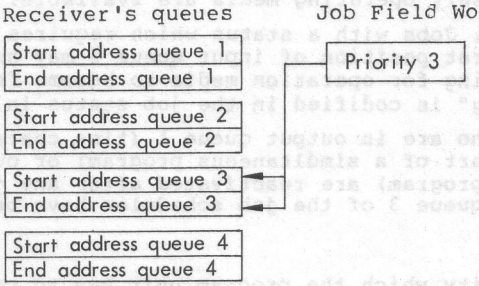
	Bit	
	1	free
	2	"1", the job is to be entered in the first position of the job scheduler's input and in the first position of the receiver's specified queue.
	3	
*1	4	Priority = the number of the input queue (-1) of the receiver in which the job field must be taken over.
	5	
	6	
	7	"1", a reply signal is stored in a cell, the address of which is given in Word 6.
	8	"1", a job field could not be connected, no time, no short time area, no program available. WAIT
	9	Assigned in the case of job field for line.
	10	
	11	"1", job field assigned.
	12	"1", job field working
	13	free
	14	= 0, jobs carried out in order = 1, no time available for program unit
	15	= 2, program unit not available = 3, no short time area available
	16	= 4, error in drive layout (false parameters) = 5, information error (e.g. read-after-write check negative
	17	= 6, device failure (e.g. ready for operation)
	18	= 7, device status message (e.g. end of paper, EOT, etc.)

*1) Priority between a program unit's jobs

Each program unit may order its jobs according to differing priorities. The quantity of priority levels depends on the receiver module. The priorities are fixed by the client.

If the client selects a priority level which the receiver module cannot handle, the job will be handled at the receiver module's lowest priority level.

Example: A job field is to be entered in the receiver program unit's third queue. The maximum quantity of queues is stipulated in Word 6 of the module supply block.



*2)

If it has not been possible to connect the jobfield, a reply signal is required in the cell, of which the address is given in Word 6. (Start address of the present User Program Index Register Label).

Jobs to be connected are entered in queue 3 of the job connector. The client may receive messages about his job by entering a code in the job status word.

In particular, the following takes place in such situations:

- Reply signal

  Bit 7 = 1 in JF Word 2: If connection has been successful, the content of the cell addressed through JF Word 6 is reduced by 1, and Bit 11 (job could be connected) is inserted in this cell.

  If a job is not connected because of a failure in the operating media such as a program unit time, or scratchpad, the corresponding failure code is entered in the job status word, the content of the cell for reply signal is reduced by one, and Bit 12 (Job could not be connected) is inserted in this cell.

  The job scheduler then removes the job field from its third input queue.

- Wait function in the job scheduler

  Bit 8 = 1 in the job field word 2: The client receives no reply signal after completion of the connection process. If the system has no time or short time area available for the receiver terminal or does not know the receiver, the job remains in the first position in input queue 3 until the necessary operating media are available.

  Exception: Jobs with a status which requires their entry in the first position of input queue 3 may supersede those jobs waiting for operation media to become free. The "cause of waiting" is codified in the job status in bits 14 to 18.

  Clients who are in output queue 1 (time characteristics: central part of a simultaneous program) or output queue 3 (central program) are reactivated after any jobs lined up in input queue 3 of the job scheduler have been completed.

- Word 3

  The activity which the program unit has to carry out is codified in this word.

  Example: 04 = positioning and/or line feed
  10 = print number of characters with control character processing
  01 = long-term job for operating the Alpha-keyboard release

- Word 4

Bit	
1	Number of the program unit which has to process the job.
2	
3	
4	
5	
6	
7	
8	
9	Number of the program unit which has created the job field.
10	
11	
12	
13	
14	
15	
16	
17	free
18	

- Word 5 (Route data)

Bit	
1	Number of the program unit which will be the next to work the job field after the first processing of the job.
2	
3	
4	
5	
6	
7	
8	
9	Gives the subsequent program unit as alternative to those in bits 1 to 8.
10	
11	
12	
13	
14	
15	
16	
17	free
18	

- Word 6

  Bits 1 to 18 give the address of a cell in which reply signals about the connected job are to be found. When job processing has finished, the content of the cell indicated by this address is reduced by one.

  Bits 11 and 12 of the cell indicated by this address are for the client to use to check job connection.

  This happens as follows (if bits 7 and 8 = 1 in word 2 of the job field).

  Bits 11 and 12 = 1 it has been possible to connect the job
  = 2 it has not been possible to connect the job, because of failure in the job status

- Words 7 to 16

  These words contain program unit specific data.

  Example: start address of a print buffer
  length of the print buffer
  position data
  address of the input code table
  address of the output code table
  etc.

- Words 17 and 18

  When the job has been carried out, error reply signals are entered in these two words.

  Word 17, program unit (device) status messages and operating errors are entered in this word.

  Word 18, device faults are entered in this word.

5 Device Control Fields

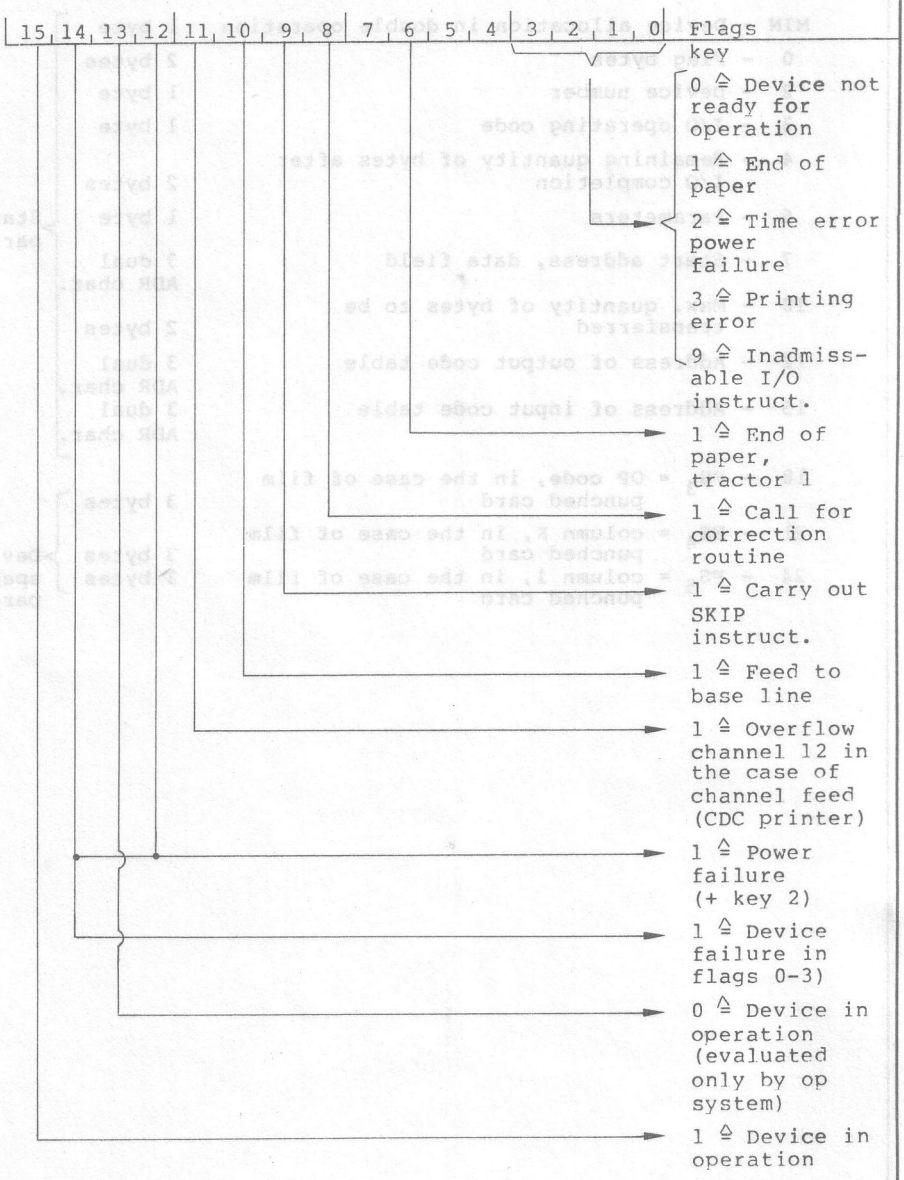
The device control fields consist of a standard part and a device-specific part. They are created in the initialisation phase, and are called by means of the table "program unit to Device Control Field", YGKFTB.

YGKFTB address: 870/4 2.4.6, 8870/6 2.0.14.

5.1 Device Control Field for Printers

MIN	- Device allocation in double operation	1 byte	Standard part = 19 B
0	- Flag Bytes	2 bytes	
2	- Device number	1 byte	
3	- I/O Operating code	1 byte	
4	- Remaining quantity of Bytes after I/O completition	2 bytes	
6	- Parameters (boundary characters)		
7	- Start address (data field)	3 dual ADR	
10	- Max. quantity of bytes to be transferred	2 bytes	
12	- Address of output code table	3 dual ADR char	
15	- Address of input code table	3 dual ADR char	
18	- Last printing position +1 after carrying out printing instruction	1 byte	Device specific part
19	- Quantity of lines for platen	1 byte	
20	- Line counter for platen	1 byte	
21	- Sheet height for roller	1 byte	
22	- Quantity of lines for tractor 1	1 byte	
23	- Line counter for tractor 1	1 byte	
24	- Sheet height for tractor 1	1 byte	
25	- Quantity of lines for tractor 2	1 byte	
26	- Line counter for tractor 2	1 byte	
27	- Sheet height for tractor 2	1 byte	
28	- Free	6 bytes	

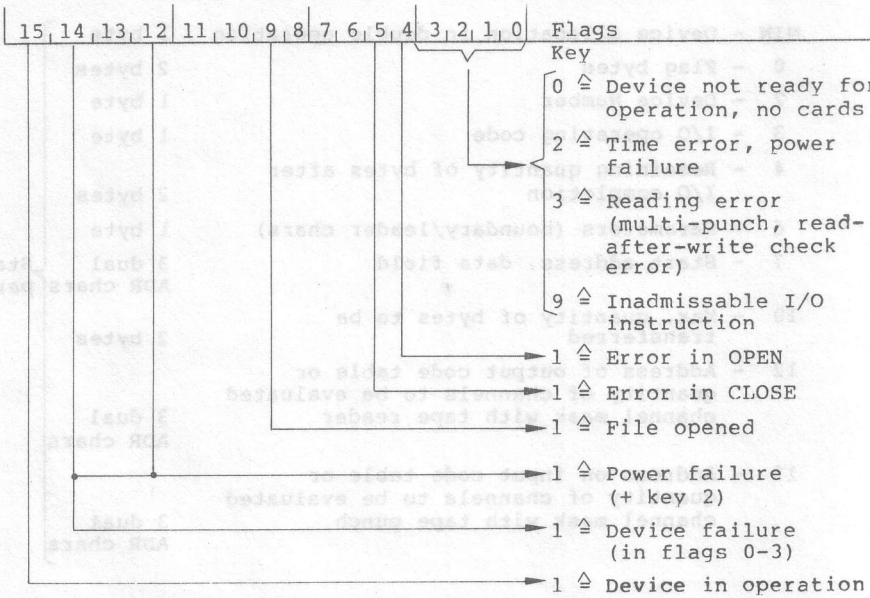
- Meaning of the Device Control Field flag bytes



5.2 Device Control Field for Punched Card

MIN	- Device allocation in double operation	1 byte	Standard part
0	- Flag bytes	2 bytes	
2	- Device number	1 byte	
3	- I/O operating code	1 byte	
4	- Remaining quantity of bytes after I/O completion	2 bytes	
6	- Parameters	1 byte	
7	- Start address, data field	3 dual ADR char.	
10	- Max. quantity of bytes to be transferred	2 bytes	
12	- Address of output code table	3 dual ADR char.	
15	- Address of input code table	3 dual ADR char.	
18	- FS_3 = OP code, in the case of film punched card	3 bytes	Device-specific part
21	- FS_4 = column K, in the case of film punched card	3 bytes	
24	- FS_5 = column 1, in the case of film punched card	3 bytes	

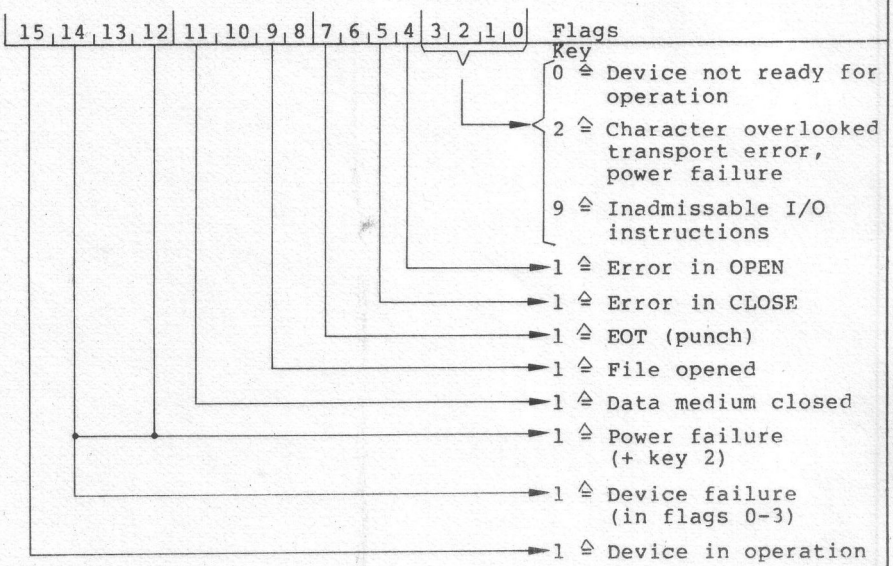
- Meaning of the Device Control Field flag bytes



5.3 Device Control Field for Punched Tape

MIN	- Device allocation in double operation	1 byte	Standard part
0	- Flag bytes	2 bytes	
2	- Device Number	1 byte	
3	- I/O operating code	1 byte	
4	- Remaining quantity of bytes after I/O completion	2 bytes	
6	- Parameters (boundary/leader chars)	1 byte	
7	- Start address, data field	3 dual ADR chars	
10	- Max. quantity of bytes to be transferred	2 bytes	
12	- Address of output code table or quantity of channels to be evaluated channel mask with tape reader	3 dual ADR chars	
15	- Address on input code table or quantity of channels to be evaluated channel mask with tape punch	3 dual ADR chars	

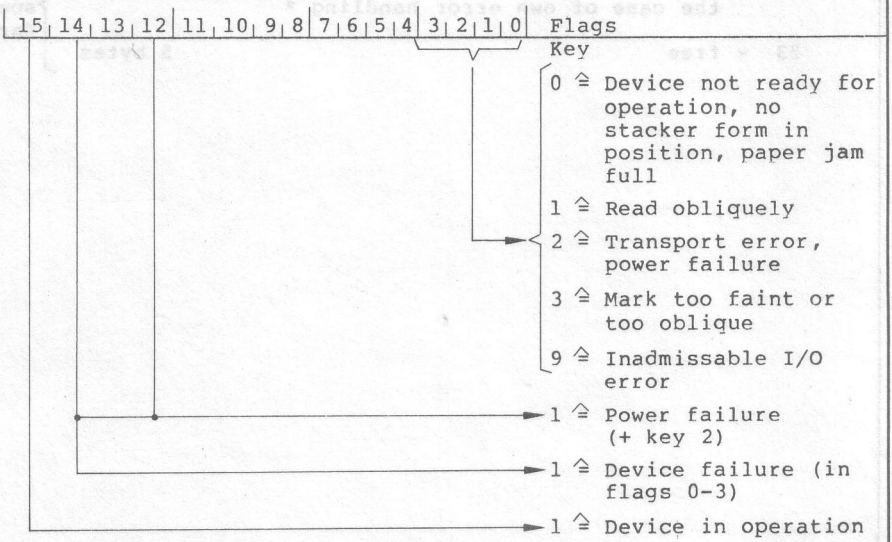
- Meaning of the Device Control Field Flag Bytes



5.4 Device Control Field for Mark Reader

MIN	- Device allocation in double operation	1 byte	Standard part
0	- Flag bytes	2 bytes	
2	- Device number	1 byte	
3	- I/O operating code	1 byte	
4	- Remaining quantity of bytes after I/O completion	2 bytes	
6	- Parameters	1 byte	
7	- Start adress, data field	3 dual ADR chars	
10	- Max. quantity of bytes to be transferred	2 bytes	
12	- Address of output code table	3 dual ADR chars	
15	- Address of input code table	3 dual ADR chars	

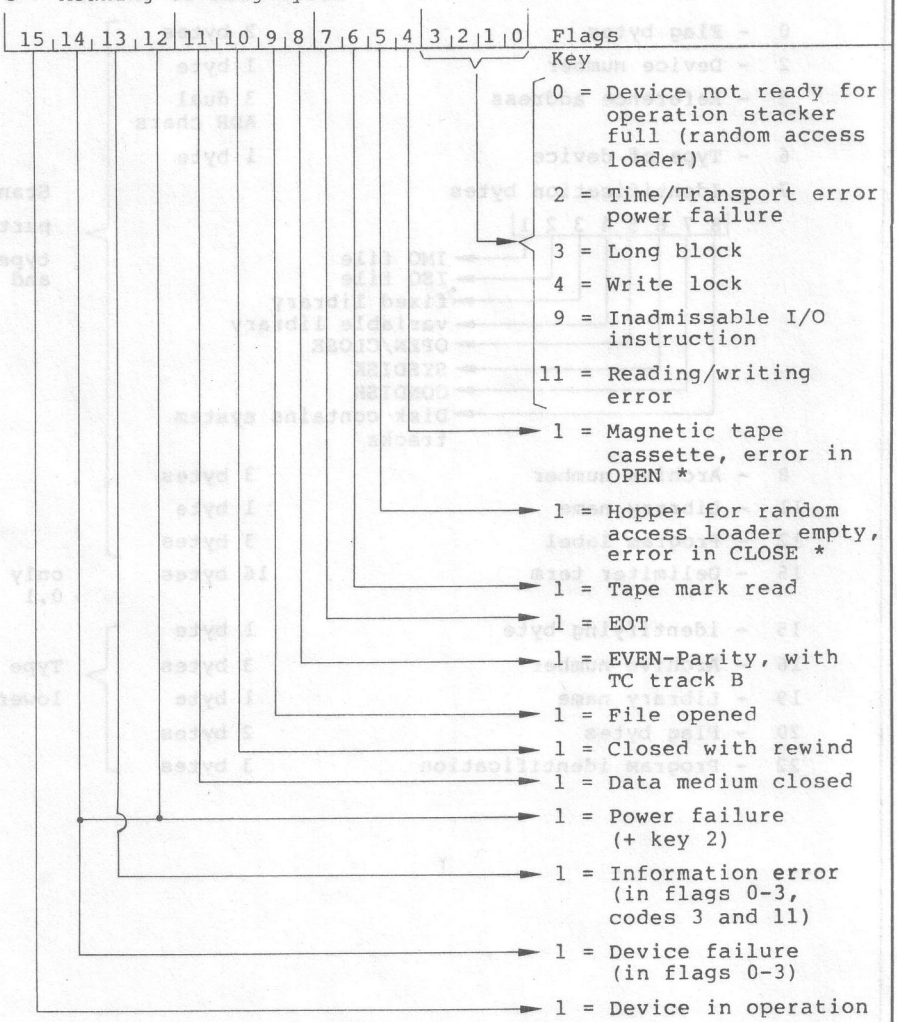
- Meaning of Flag Bytes



5.5 Device Control Field for Magnetic Tape/Magnetic Tape Cassette

MIN	- Device allocation in double operation	1 byte	Standard Part
0	- Flag bytes	2 bytes	
2	- Device number	1 byte	
3	- I/O operating code	1 byte	
4	- Remaining quantity of bytes after I/O completion	2 bytes	
6	- Parameters (tape marking code)	1 byte	
7	- Start address , data field	3 dual ADR chars	
10	- Max. quantity of bytes to be transferred	2 bytes	
12	- Address of output code table	3 dual ADR chars	
15	- Address of input code table	3 dual ADR chars	
18	- Date of last use (decimally packed)	4 bytes	Device-specific Part
22	- Flag byte 1 (bits 5 and 6) doubled in the case of own error handling *	1 byte	
23	- free	5 bytes	

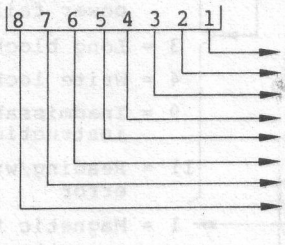
- Meaning of Flag Bytes



* In the case of error specified in IOCS, PRG 17, errors in OPEN/CLOSE are indicated in flags 4 and 5. The old flag setting of Flags 4 and 5 are in MDEV, DN 22.

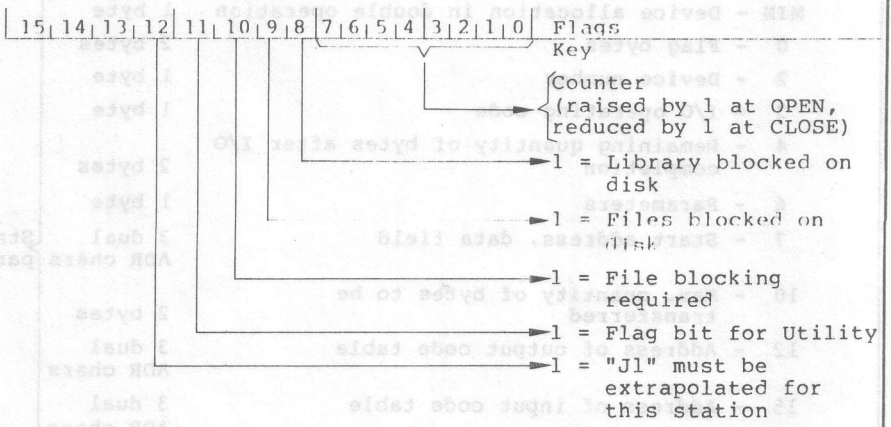
5.6 Device Control Field for Magnet Disk

0	- Flag bytes	2 bytes	Standard part for types 0,1 and 3
2	- Device number	1 byte	
3	- Reference address	3 dual ADR chars	
6	- Type of device	1 byte	
7	- Identification bytes		



8	- Archive number	3 bytes	
11	- Library name	1 byte	
12	- Program label	3 bytes	
15	- Delimiter term	16 bytes	only types 0,1
15	- identifying byte	1 byte	Type 3, lower half
16	- Archive number	3 bytes	
19	- Library name	1 byte	
20	- Flag bytes	2 bytes	
22	- Program identification	3 bytes	

- Meaning of flag bytes 0 and 1



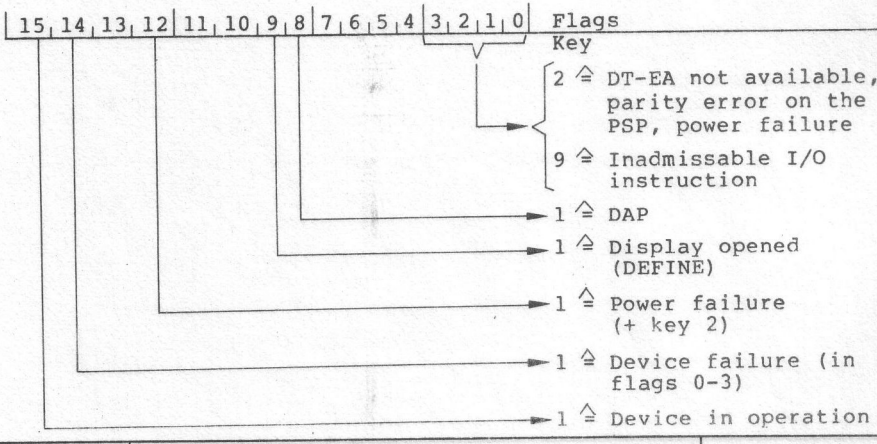
- Meaning of bytes 12, 13 and 14 (program label)

Byte 12 = (SYX, X61) Utility Flag bit 11
Byte 13 = (SYX, X61) by "files blocking", bit 9
Byte 14 = (SYX, X61) by "library blocking", bit 8

5.7 Device Control Field for Display

MIN	- Device allocation in double operation	1 byte	Standard part
0	- Flag bytes	2 bytes	
2	- Device number	1 byte	
3	- I/O operating code	1 byte	
4	- Remaining quantity of bytes after I/O completion	2 bytes	
6	- Parameters	1 byte	
7	- Start address, data field	3 dual ADR chars	
10	- Max. quantity of bytes to be transferred	2 bytes	
12	- Address of output code table	3 dual ADR chars	
15	- Address of input code table	3 dual ADR chars	
18	- FS_3 character	3 bytes	Device-specific Part
21	- FS_4 character, screen buffer address/ start address of the 6 special bytes	3 bytes	
24	- Absolute screen size	2 bytes	
26	- Lock key setting in bits 1 to 4	1 byte	
27	- Intermediate buffer for keyboard inputs	15 bytes	Only in the case of DAP

- Meaning of the Flag bytes



5.8 Line Device Control Field

MIN - Line program allocation in double operation — 1 byte

0 - OPEN STATUS — 1 byte

00 = no OPEN; CLOSE has been performed
01 = CLOSE working
80 = OPEN has been performed
40 = it has not been possible to perform OPEN

1 - OPEN STATUS — 1 byte

01 = OPEN working

2 - Device number — 1 byte

3 - Failure byte — 1 byte

4 - Failure Program address — 2 bytes (16 bit)

- Meaning of the bits in the line failure byte

7	6	5	4	3	2	1	0	Flags

Key

1 ≙ Stop line processing (only for test purposes; does not bring about any internal failure)

2 ≙ Illegal instruction sequence

3 ≙ Buffer header byte not cleared

4 ≙ OPEN not present for line

5 ≙ Incorrect instruction arrangement

6 ≙ Incorrect program unit control field arrangement

7 ≙ Inadmissable address data in program unit control field or instruction

5.9 Program Unit Control Field of Transmission Line

The terminal control fields are created by the user. Their start addresses are to be found in the allocated present line instructions (M-addr).

Rel. ADR	Length (bytes)	Description	Meaning
0	1	TMK	With Nixdorf daisy chain procedure, flag register can contain one of the FNR sent from the TCU.
1	2 (bin)	TJTIME	Job limit time for READ/ WRITE instructions, (min 1 s; = 0 = no time limitation)
3	1 (bin)	TTNL	Program unit number on the line
4	3 (dual)	TDAE	Start address of the receiver buffer
7	3 (dual)	TAAE	Start address of the working buffer* (for GET instructions)
10	2 (bin)	TDLE	Block length in the receiver buffer (max. 65535)
12	1 (bin)	TMB	Quantity of sub-buffers in the multi-block buffer *
13	3 (dual)	TDAS	Start address of the send buffer
1.0	2 (bin)	TDLS	Length of the send buffer (max. 65535)
Variant B (UP Word 10 bit 9 = 0)			
1.2	3 (dual)	THDRL	Start address of the header buffer (for WRITE and OPENAW instructions)
1.5	1	THDRL	Length of the header data
1.6	2 (bin)	TITBL	Length until ITB is inserted in data block
Variant A (UP Word 10 bit 9 = 1) The following data are not evaluated, but are controlled by the UP.			
1.2	3 (dual)	THDRP	Start address of the (identifiers and addresses are the same)
	1	TEZ	End character of program unit control field (15.15)

Only required for LDA 900. TIOCS 901 uses these bytes differently.

→ Fixed terminal control field arrangement.
→ Variable terminal control field arrangement.

5.10 BDC Line Job Control Field

When a Get-All instruction is pending, the reply signals from the job field are filed in the job control field after a job has been fully performed (line instruction carried out).

The job control field start address is to be found in the Get-all instruction M-data.

Rel. ADR	quantitiy of bytes	Description	Message	
+ 0	1	RGNR	Line no. (device no.)	
+ 1	1	RTNL	Terminal line	
+ 2	1	ROP	OP-code (instruction FS 2 code)	
+ 3	1	RPROZ	Procedure reply signals	
+ 4	1	RFCODE	Error code	error signal
+ 5	1	RFEEA	Last I/O status	
+ 6	2	RFEADR	Error address 16 bits.	
+ 8	3	RFTIME	Seconds	
+ 11	3	RTKF	Program unit control field address	
+ 14	3	RPANF	Buffer address	
+ 17	2	RZAHL	Quantity of data characters	
+ 19	1	RHDRL	Length of header data	

If an error signal is waiting in RFCODE, Bit 8 = 1 is inserted in reply signal store RPROZ.

For notes

6 Software Bell

The software bell is used for the simultaneous time management of the system. It is called up via bell queue YWKAND.

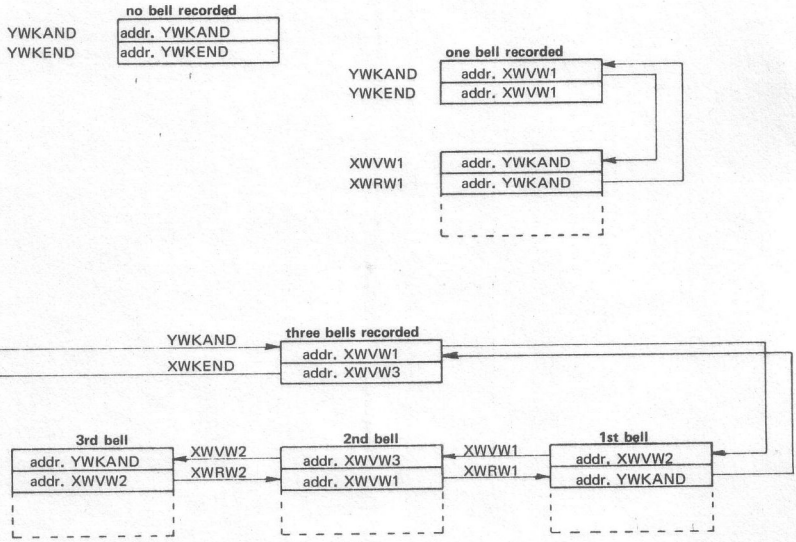
YWKAND addr. 8870/4 2.1.C, 8870/6 2.0.0.

	18	1 Bit
1st word	XWVW,	Forward reference
2nd word	XWRW,	Backward reference
3rd word	XWKZI,	Scratchpad area refererence
4th word	XWWDH,	Repetition counter
5th word	XWLFD,	Running counter

Since all simultaneously operating program units can activate bells, there is a queue for the bells. They are chained to one another through Words XWVW, while the address of the previously activated bell is recorded in XWRW.

The bell queue consists of the two memory words YWKAND and YWKEND. The XWVW address of the first bell activated is to be found in YWKAND, while the XWRW address of the last recorded bell is to be found in YWKEND. If no bell has been recorded, the YWKAND address is to be found YWKAND and YWKEND.

- Bell Queue



- Word 3

  The start address of the program unit's short time area to which the bell is allocated is to be found in XWKZI.

- Word 4

  If the bell is to operate cyclically, its finishing time is to be found in XWWDH, bits 2 to 18. If XWWDH Bits 2 to 18 = 0, there is no cyclical bell, i.e. the bell only goes off once.

  Bit 1 determines whether the program unit micro is to be called via Instruction Counter 1 or Instruction Counter 2.

  If Bit 1 = 0, it is called via Instruction Counter 1
  (Short time Word 5)
  If Bit 1 = 1, it is called via Instruction Counter 2
  (Short time Word 6)

- Word 5

  The time during which the bell is to be active is given in units of 4 ms at the beginning of XWLFD. XWLFD is reduced by 1 every 4 ms.

  If XWLFD = 0, the program unit is called.

7 Equipment and System Parameters

The equipment parameters contain the current information for the program unit (bell and I/O interrupt numbers, device number, etc.), while the system parameters contain system-specific data such as quantity of job fields and dynamic short time areas, and interrupt numbers for core and DMA units.

These parameters are required in the initialisation phase for the apportionment of the working memory and system part.

There is a module supply block consisting of three words in front of the equipment and system parameters for purposes of identification.

- Word 1

  Contains the search argument of the equipment and system parameter module.

  If Word 1 reads 0.0.0.0.15, it concerns the equipment and system parameters for Systool (Block 3, 12).

  If Word 1 reads 0.0.0.0.14, it concerns the equipment and system parameters for the overall system (full initialisation in Block 3, 11).

- Word 2

  Gives the total length of the equipment and system parameters in the number of words.

- Word 3

  Specifies that the equipment and system parameters consist of only one module and may have no submodules.

18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit
word with content "14" = module supply block for the equipment and system parameters																		This module supply block consists of only 3 words. No submodules allowed.
module length (equipment parameters + system parameters)																		
									1								1	
		spec. funct.	module number											call bit	time characteristics			Equipment parameter word 1 (identical to module supply block word 1.)
		program unit number								priority in bits 1-8 for centr. progr. units otherwise bits 1-4				I/O INT1	I/O INT2	bell INT2	bell INT1	
		bell number (instruct. counter 1)																
		bell number (instruct. counter 2)																
										I/O interrupt no. (instruction counter 1)								If required in this sequence.
										I/O interrupt no. (instruction counter 2)								
		free for further sequence parameters			2nd dev. no.	ig. of DCF	1st dev. no.	load no.	DMA no.	number of sequence param. to be taken over into the long-time area								Equipment parameter index
		Sequence parameters which to be taken over into the long-time area.																
		logical DMA no. (replaces word 6 of the module supply block)																In this sequence, if given in the index.
		load number											load number for time requirem. on activation					
		first device number																
		length of device control field in bytes																
		second device number (must be first device number)																
		word = zero = end of equipment parameters																Either word 1 of the next equipment param. record or end of the eqpt. param.
		quantity of additional global cells to be kept free																System parameter word 1 Variable quantity of param. max. 32 words for 15 short-time classes + job fields. Short time class 0 = JF = last parameter.
		length of a byte area to be kept free for additional device control fields																
		short time class	quantity of short time areas to be created for class to be entered in bits 13-16, or quantity of job fields to be created of there is a 0 in bits 13-16															
		length of a short time area for previously entered class or length of a job field																
		I/O address																System output table The table consists of word pairs: 1st word = I/O addr. 2nd word = issue value The table is also run through after power fail. 1st word = I/O addr. = zero = end of table.
		value to be issued																
		I/O address																
		value to be issued																
		zero = end of the table																
		first word of the device combination table																Device combination table. Length = variable.
		zero = end of the system parameters																

7.1 Installation Parameters

- Word 1 (first word of the equipment parameters)

Bit	
1	Not used
2	= 0, the program unit specified in word 2, bits 9 to 16, is operating centrally
3	= 2, the program unit specified in word 2, bits 9 to 16, is operating simultaneously and cannot be interr.
4	= 1, the module to be called by the Call Instruction (change UP to Operating Program).
5	
6	
7	
8	Gives the module number
9	
10	
11	
12	
13	
14	
15	
16	= 1, for this module, a job instruction may be generated on the UP level (JOBAB instruction)
17	not used
18	

- Word 2

Bit	
1	= 1, the program unit specified in bits 9 to 16 is working with bell interrupt 1. The number is in word 3 (bits 9 to 16)
2	= 1, the program unit specified in bits 9 to 16 is working with bell interrupt 2. The number is in word 4 (bits 9 to 16)
3	= 1, the program unit specified in bits 9 to 16 is working with I/O interrupt 1. The number is in word 5 (bits 1 to 6)
4	= 1, the program unit specified in bits 9 to 16 is working with I/O interrupt 2. The number is in word 6 (bits 1 to 6)
5 6 7 8	If the program unit specified in bits 9 to 16 is operating centrally, bits 1 to 8 specify the priority with which its jobs are to be entered in output queue 3 of the job scheduler. Bits 1 to 8 = 0 = highest priority.
9 10 11 12 13 14 15 16	Program unit number allocated to the module specified in word 1.
17 18	free

- Word 3 (applies to simultaneously operating program units)*

Bit	
1	free
2	
3	
4	
5	
6	
7	
8	
9	Gives information of the bell number if Bit 1 is inserted in Word 2. This number is allocated to the interrupt reference address in short time word 5 (instruction counter 1).
10	
11	
12	
13	
14	
15	
16	
17	free
18	

* If the program unit is operating centrally, Word 3 already contains the index.

- Word 4 (applies to simultaneously operating program units)

Bit	
1	free
2	
3	
4	
5	
6	
7	
8	
9	Gives information on the bell number if Bit 2 is inserted in Word 2. This number is allocated to the interrupt reference address in short time word 6 (instruction counter 2).
10	
11	
12	
13	
14	
15	
16	
17	free
18	

- Word 5 (applies to simultaneously operating program units)

Bit	
1	Gives information on I/O interrupt 1 if Bit 3 is inserted in Word 2. The interrupt no. is allocated to the interrupt reference address through short time word 5 (instruction counter 1).
2	
3	
4	
5	
6	
7	
8	
9	free
10	
11	
12	
13	
14	
15	
16	
17	
18	

- Word 6 (applies to simultaneously operating program units)

Bit	
1	Gives information o I/O interrupt 2 if Bit 4 is inserted in Word 2. The interrupt number is allocated to the interrupt reference address through short time word 6 (instruction counter 2).
2	
3	
4	
5	
6	
7	
8	
9	free
10	
11	
12	
13	
14	
15	
16	
17	
18	

- Word 7 (index of further parameters)

Bit	
1	Gives the number of sequence parameters from the following word for the program units specified in word 2 (bits 9 to 16) For example: I/O disk select address, start and end address of the cyclical buffer (in the case of needle printer) or output address of the DMA and DMA feature or start address of the print character output table, etc. This data is transferred into the long-time area.
2	
3	
4	
5	
6	
7	= 1, the program unit is working with DMA, i.e. this program units logical XX DMA no. is in the first word after the sequence parameters.
8	= 1, a load number has been allocated to the program units (the program unit is operating simultaneously). The load number is to be found in the subsequent data.
9	= 1, a device number has been allocated to the program unit.
10	= 1, a device control field has been allocated to the program unit; its length will be shown in the subsequent parameter data.
11	= 1, a second device number has been allocated to the program unit
12	
13	Left free for further parameters.
14	
15	
16	
17	
18	

- Word 8

  The sequence parameters of which the quantity is specified in the index (bits 1 to 6) are given from this word onwards.

- Word n (word after the sequence parameters)

  Contains the numbers of the DMA's with which the program unit works.

- Word n + 1

  Contains the system load from which the program unit originates in the system when it is called in.

- Word n + 2

  Contains the device number referred to through Bit 9 of the index.

- Word n + 3

  Contains the length in bytes of the device control field referred to through Bit 10 of the index.

- Word n + 4

  Contains the second device number, referred to through Bit 11 of the index. This number must be smaller than the first device number.

- Word n + 5

  This is either word 1 of the next installation parameter label or, if its content = 0, indicated the end of the equipment parameters.

7.2 System Parameters

The equipment parameters are generally followed by the system parameters. They are separated by a single word, the content of which = 0.

- Word 1

  This contains the number of additional memory cells, not normally allowed for in the standard version of the system, which have to be kept free. (Reserved on initialisation).

- Word 2

  This contains the number of additional bytes not normally allowed for in the standard version of the system, which have to be kept free. (Reserved on initialisation).

- Word 3

Bit	
1	States how many times the first dynamic short time area is to be created.
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	States how the first short time class is to be recognised.
14	
15	
16	
17	free
18	

- Word 4

  Gives information on the length of the first dynamic short time area for the class specified in word 3 (bits 13 to 16).

- Word 5

Bit	
1	States how often the second dynamic short time area is to be created.
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	States how the second dynamic short time class is to be recognized.
14	
15	
16	
17	free
18	

- Word 6

  Gives information of the length of the second dynamic short time area for the class entered in word 5 (bits 13 to 16). Up to a maximum of 15 further short time classes may be given through word pairs.

- Word n (first word after the word pairs in the short time class information)

Bit	
1	States how often a job field is to be created.
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	These bits are zero, that is to say the information in bits 1 to 8 applies to a job field and not to a short time area.
14	
15	
16	
17	free
18	

- Word n + 1

  Gives information on the length of a jobfield. Information on how often the job field is to be created is given in the preceding word.

The system output table follows from word n + 2 onwards. In this, two words represent each value.

- Word n + 2

  Contains an address for an I/O unit.

- Word n + 3

  Gives the output code for the I/O address given above.

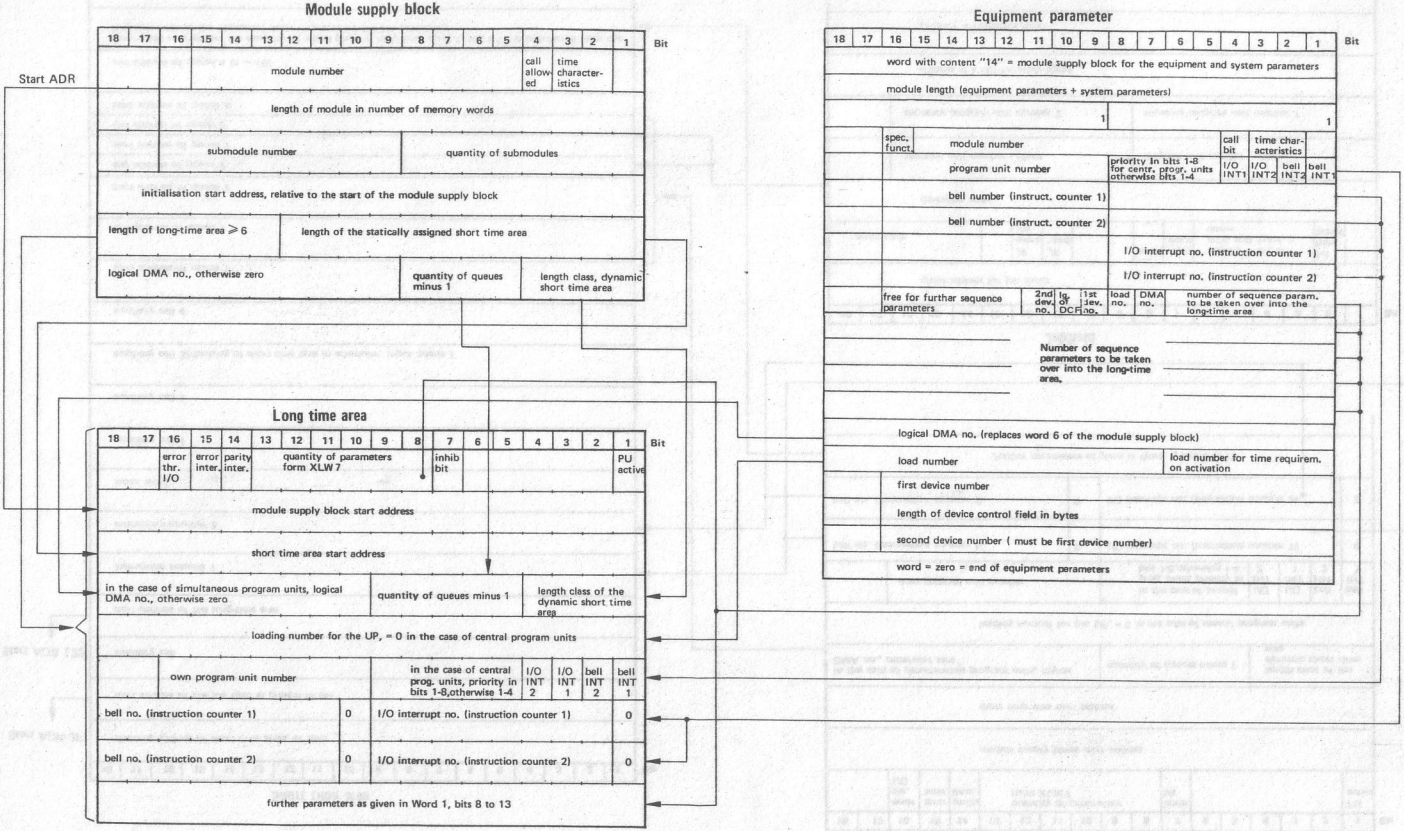
The system output table is followed for the I/O address given above.

This table contains data on which combinations may operate simultaneously in the device.

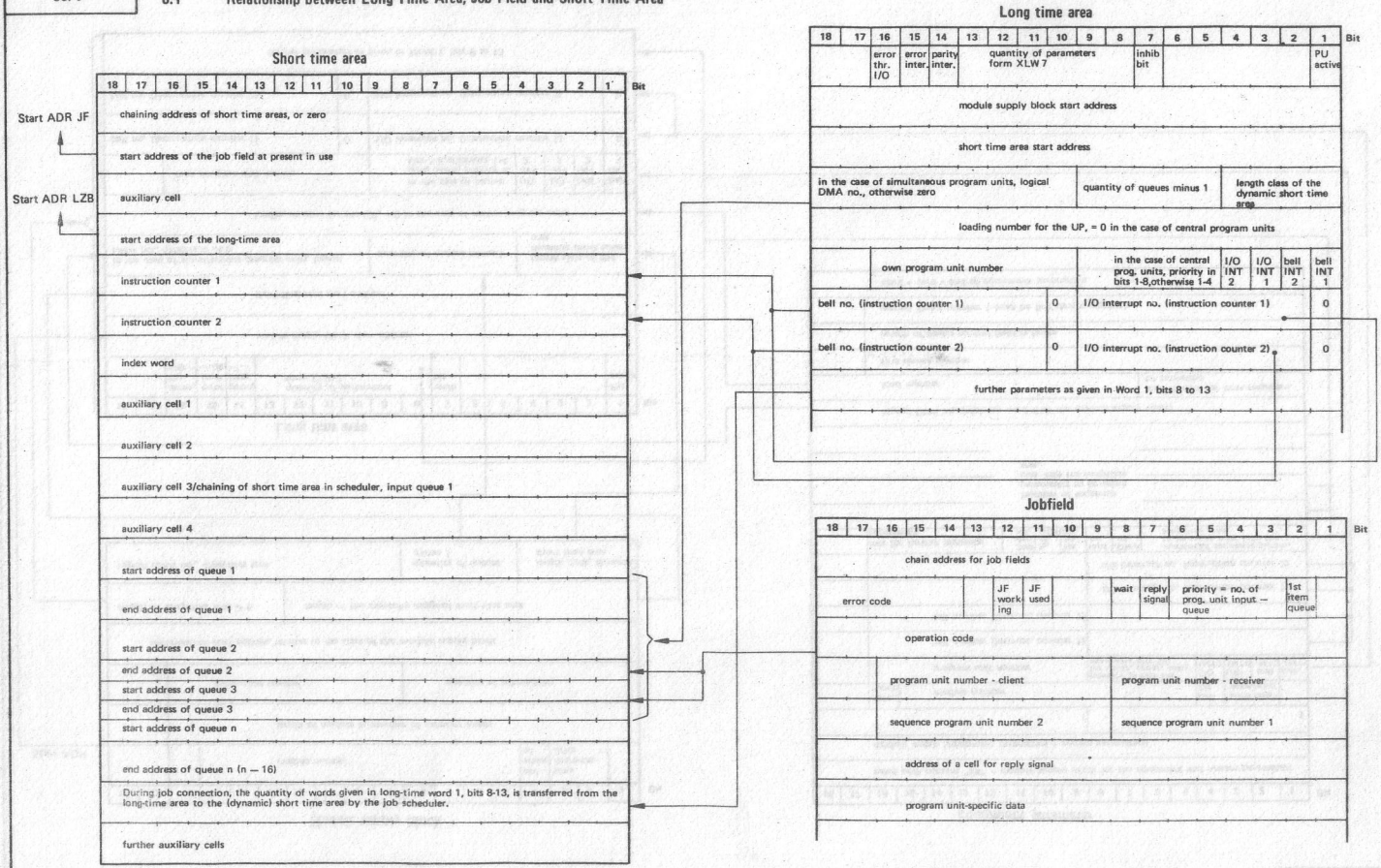
The device combination table is terminated by a word containing zero. This simultaneously indicates the end of the equipment and system parameters.

At present this table is not used.

For notes


Module supply block
Start ADR
Bit
module number
call allowed
time characteristics
length of module in number of memory words
submodule number
quantity of submodules
initialisation start address, relative to the start of the module supply block
length of long-time area ≥ 6
length of the statically assigned short time area
logical DMA no., otherwise zero
quantity of queues minus 1
length class, dynamic short time area
Long time area
error thr. I/O
error inter.
parity inter.
quantity of parameters form XLW 7
inhib bit
PU active
module supply block start address
short time area start address
in the case of simultaneous program units, logical DMA no., otherwise zero
quantity of queues minus 1
length class of the dynamic short time area
loading number for the UP, = 0 in the case of central program units
own program unit number
in the case of central prog. units, priority in bits 1-8, otherwise 1-4
I/O INT 2
I/O INT 1
bell INT 2
bell INT 1
bell no. (instruction counter 1)
I/O interrupt no. (instruction counter 1)
bell no. (instruction counter 2)
I/O interrupt no. (instruction counter 2)
further parameters as given in Word 1, bits 8 to 13
Equipment parameter
word with content "14" = module supply block for the equipment and system parameters
module length (equipment parameters + system parameters)
spec. funct.
module number
call bit
time characteristics
program unit number
priority in bits 1-8 for centr. progr. units otherwise bits 1-4
bell number (instruct. counter 1)
bell number (instruct. counter 2)
I/O interrupt no. (instruction counter 1)
I/O interrupt no. (instruction counter 2)
free for further sequence parameters
2nd dev. no.
lg. of DCF
1st dev. no.
load no.
DMA no.
number of sequence param. to be taken over into the long-time area
Number of sequence parameters to be taken over into the long-time area
logical DMA no. (replaces word 6 of the module supply block)
load number
load number for time requirem. on activation
first device number
length of device control field in bytes
second device number (must be first device number)
word = zero = end of equipment parameters

8.1 Relationship between Long-Time Area, Job Field and Short Time Area



9 Working Memory Assignment

The working memory is assigned on initialisation.

9.1 Assignment in the case of 8870/2/4

SC Area	0 - 1.14	Hardware and diagnosis cells.
	0.2.0 ...	I/O Stackpointer (max. 240 cells)
		Long-time and static short time areas, working area of initial loader.
	2.0.0 - 2. 7.14	Global cells ("Y" cells).
	2.8.0 - 2.13.14	Job scheduler cells initialisation shift cells, and bell working cells.
	2.14.0	Additional global cells according to installation parameters.
	YSYSPAR (2.4.2)	System output table.
	XVKOTOB (2.4.4)	Device combination table (occupies only one cell, as this table is not used).
	YGKFTB (2.4.6)	"Program unit to Device Control Field" table (max. 256 cells).
	YTNLTB (2.4.8)	"Program unit to Long-Time Area" table (max. 256 cells).
	YWTAB (2.5.2)	"Bell number to short time area" table (largest bell number times (5) cells).
	YINKTB	"Interrupt no. to short time area" table (max. 240 cells).
	YGERTN (2.4.12)	"Device number ot program unit number" table (128 cells). (1 place in table = 8 bits 1 cell = 2 places in the table).
YGTEND (2.4.14)		Further long-time and static areas (index and system registers).
		Dynamic short time areas.
		Job fields

Lower Half of Main Memory Area

YHSSAS (2.5.6)	Start of byte area	Input area keyboard for single or foreground (83 bytes).
		Input area keyboard for background (83 bytes).
		Parameter field for display (18 bytes).
		2 x screen buffer (2 x 960 or 1920 bytes).
YHSDAS (2.5.8)		Device control fields.
		Common area (326 bytes)
XDAS1 (X055)		System working area (1088 bytes)
		User area
HSEND (2.5.A)		

9.2 Assignment in the case of 8870/6

The working memory is assigned on initialisation.

These cells are found as working cells in CPU 1501	0. 0. 0 - 0. 1.14	Hardware and diagnosis cells
	0. 2. 0 - 0. 3.14	Table fo register labels (SATZTAB). These contain the start address of the oriented programs register records of transaction.
	0. 4. 0 - 0. 9.14	Working cells for the job scheduler and the interrupt processing.
	0.10. 0 - 0.11.14	Working cells for CPU 1501 (internal).
	0.12. 0 - 0.15.14	Working cells of the macro-interpreter and the bell management.
	1. 0. 0 - 1. 7.14	System register of the present program unit.
	1. 8. 0 - 1.15.14	Index register of the present program unit.
	2. 0. 0 - 2. 8.10	Y global cells (general system working cells).
	2. 8.12 - 2.10.10	Global cells for LTG-IOCS. (Working cells for BDC line control).
	2.10.12 - 3.14.14	Working cells for initial loader.
	3.15. 0 ...	Additional global cells according equipment parameters.
	YSYPAR (2.0.10)	System output table.
	YVKOTB (2.0.12)	Device combination table (not used at present).
	YGKFTB (2.0.14)	"Program unit to Device Control Field" table.
	YTNLTB (2.1.0)	"Program unit to Long-Time Area" table.
	YWTAB (2.1.8)	"Bell number to short time area" table. (Bell no. times 5 cells, 5 cells per bell).
	YINKTB (0.4.4)	"Interrupt Number to short time area" table
	YGERTN (2.1.4)	"Device number to program unit number" table.

YGTEND (2.1.6)		Macro-interpreter static short time area (13 cells).
		System register (fore-or background).
		Index register (fore-or background).
		Further long-time areas and static short time areas.
		Dynamic short time areas.
YHSSAS (0.13.12)		Job fields
XSAS (X 32 S)	Byte area	Input buffer keyboard, foreground or single (83 bytes)
		Input buffer keyboard, background (83 bytes)
		Parameter field for display (18 bytes)
		System display buffer (960 or 1920 bytes).
		Working display buffer (960 or 1920 bytes).
YHSDAS (2.1.14)		Device control fields
		Common area (326 bytes)
YDAS1 (X05S)		System working area (1088 bytes)
		Beginning of the free user area.
YHSEND (2.2.0)		

10 Reference and Working Table in the Working Memory

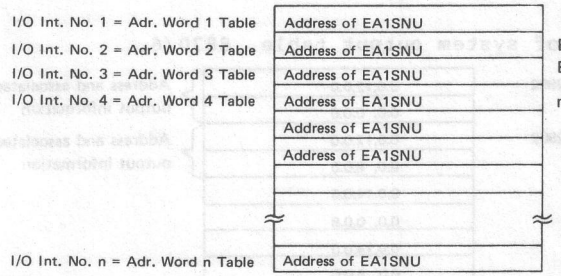
Reference tables are used to locate memory areas such as long-time areas, short time areas, device control fields, and job fields for the program unit, while working tables contain data required from time to time by the system, such as system output tables, device combination tables, and bit assignment tables.

10.1 Reference Tables

10.1.1 I/O Stackpointer (only in the case of 8870/2/4)

The I/O interrupt number is allocated to the entry address of the I/O interrupt routine via the stack pointer. There are two entry addresses, EA1SNU and EA2SNU, depending on whether the I/O interrupt number is allocated to instruction counter 1 or 2.

- Table from 0.2.0: I/O stack pointer



An "I/O interrupt no. to I/O interrupt routine entry address" reference table is not necessary in the case of 8870/6, as interrupt evaluation is carried out in CPU 1501.

10.1.2 System Output Table

This table is required on initialisation and when re-starting after power failure. The data in the table is identical to that in the system output table in the system parameters.

The table is called via global cell YSYPAR address YSYPSR, 8870/4 2.4.2, 8870/6 2.0.10.

- Assignment of system output table 8870/4.

Coupling unit address 2802	0.0.12.0.0	Address and associated output information
Interrupt no. 2802	0.0. 0.1.0	
Coupling unit address 2802	0.0.12.0.0	Address and associated output information
Interrupt release	0.0. 8.0.0	
I/O address 1802	0.0.10.0.0	
Interrupt no. 1802	0.0. 0.1.1	
I/O address 1802	0.0.10.0.0	
Interrupt release 1802	0.0. 8.0.0	
End of table	0.0. 0.0.0	

- Assignment of system output table, 8870/6.

Coupling unit address 2802	0.0.12.0.0	Address and associated output information
Interrupt no. 2802	0.0. 0.0.0	
Coupling unit address 2802	0.0.12.0.0	Address and associated output information
Interrupt release 2802	0.0. 8.0.0	
DMA address 1804	0.0.14.0.0	
Interrupt no. 1804	0.0. 0.0.8	
DMA address 1804	0.0.14.0.0	
Interrupt release 1804	0.0. 8.0.0	
I/O address 1802	0.0.10.0.0	
Interrupt no. 1802	0.0. 0.0.1	
I/O address 1802	0.0.10.0.0	
Interrupt release	0.0. 8.0.0	
End of table	0.0. 0.0.0	

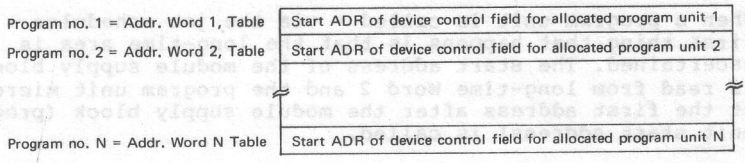
10.1.3 Device Combination Table

At the moment, this table is not used. All the peripheral devices at present connected to the computer can interact with one another.

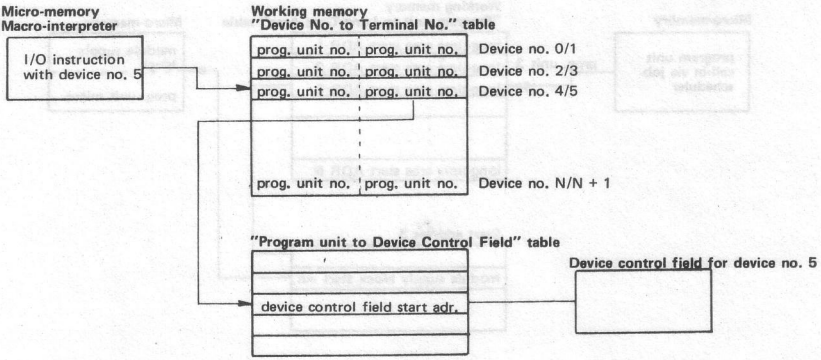
10.1.4 "Program Unit to Device Control Field" Table

This table is used to ascertain the start addresses of the device control fields by first giving the program unit number. It is used when, in the case of XX I/O instructions, the device control field address must be ascertained via the device number.

The start number is in YGKFTB ADR 8870/4 2.4.6, 8870/6 2.0.14



10.1.4.1 Relationship between the Device/Program Unit Number and the Device Control Field



10.1.5 Program Unit to Long-Time Area

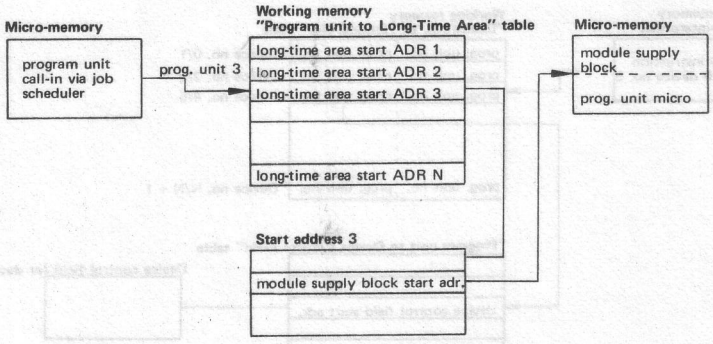
This table serves to allocate the long-time area to the program unit number. It is used when calling a program unit (i.e. activating it).

Start address is YTNLTB ADR 8870/4 2.4.8, 8870/6 2.1.0

Program unit no. 1 = Addr. Word 1 Table	Start ADR of long-time area for allocated program unit 1
Program unit no. 2 = Addr. Word 2 Table	Start ADR of long-time area for allocated program unit 2
Program unit no. 3 = Addr. Word 3 Table	Start ADR of long-time area for allocated program unit 3
Program unit no. N = Addr. Word N Table	Start ADR of long-time area for allocated program unit N

When a program unit is called - via the job scheduler - the first thing that happens is that the long-time area is ascertained. The start address of the module supply block is read from long-time Word 2 and the program unit micro at the first address after the module supply block (program unit start address) is called.

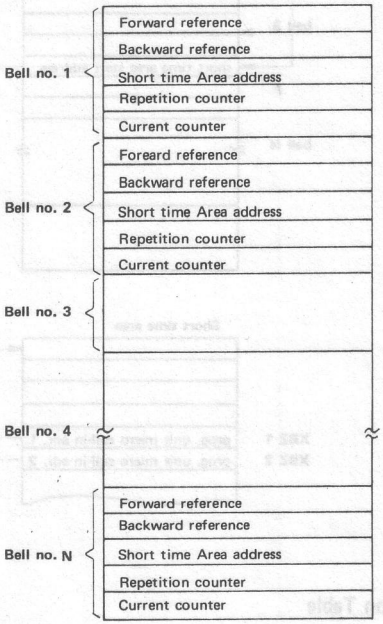
10.1.5.1 Relationship between Program Unit, Long-Time Area, Module Supply Block and Program Unit Micro



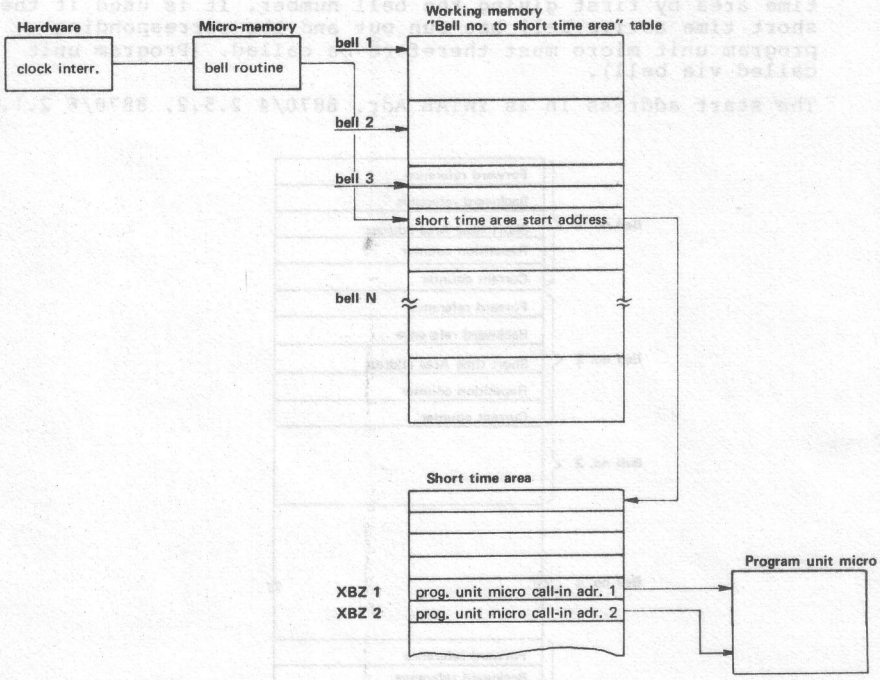
10.1.6 "Bell Number to Short Time Area" Table

This table is used to locate the start address of the short time area by first giving the bell number. It is used if the short time active bell has run out and the corresponding program unit micro must therefore be called. (Program unit called via bell).

The start address in is YWTAB Adr. 8870/4 2.5.2, 8870/6 2.1.8



10.1.6.1 Relationship between Clock Interrupt, Bell Table and Program Unit Micro



10.1.7 Device Combination Table

At present this table is not used. All the peripheral devices connected to the computer can interact with one another.

10.1.8 "Program Unit to Device Control Field" Table

This table is used to ascertain the start address of the device control field via the program unit number.

The start address is in YGKFTB Adr. 8870/4 2.4.6, 8870/6 2.0.14

Program unit no. 1 = Addr. Word 1 tab.	Start ADR of device control field for allocated progr. unit 1
Program unit no. 2 = Addr. Word 2 tab.	Start ADR of device control field for allocated progr. unit 2
Program unit no. 3 = Addr. Word 3 tab.	Start ADR of device control field for allocated progr. unit 3
Program unit no. N = Addr. Word N tab.	Start ADR of device control field for allocated progr. unit N

10.1.9 Interrupt Number to Short Time Area

This table is used to allocate the short time area to the corresponding interrupt number.

The start address of the table is in YJNKTB.

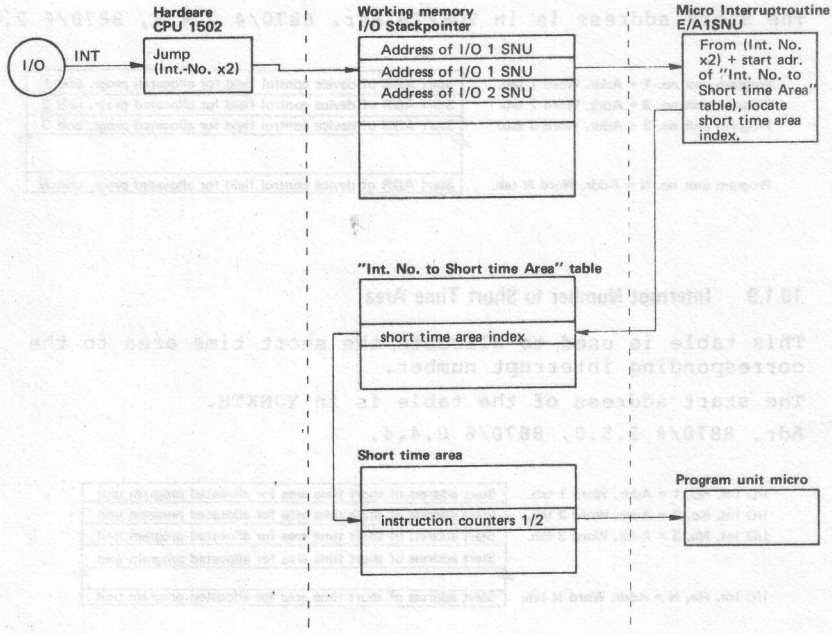
Adr. 8870/4 2.5.0, 8870/6 0.4.4.

I/O Int. No. 1 = Addr. Word 1 tab.	Start address of short time area for allocated program unit
I/O Int. No. 2 = Addr. Word 2 tab.	Start address of short time area for allocated program unit
I/O Int. No. 3 = Addr. Word 3 tab.	Start address of short time area for allocated program unit
	Start address of short time area for allocated program unit
I/O Int. No. N = Addr. Word N tab.	Start address of short time area for allocated program unit

If the program units are not active, the interrupt end routine entry is present in the allocated place in the table. (Short circuit against undesired interrupts).

If a program unit is activated, the start address of the short time area is entered in the corresponding place in the table.

10.1.9.1 Relationship between I/O Interrupt and I/O Stackpointer



11 "Device Number to Program Unit Number" Table

This table is used to allocate the program unit number to the preceding device number. It is used when an I/O instruction is inserted and the associated device control field must be located.

The start address of the table is in YGERTN, Adr. 8870/4 2.4.12, 8870/6 2.1.4.

Two device numbers are allocated to each place in the table.

	18 10	9 1
Device No. 0/1	Program unit No.	Program unit No.
Device No. 2/3	Program unit No.	Program unit No.
Device No. 4/5	Program unit No.	Program unit No.
Device No. 6/7	Program unit No.	Program unit No.
Device No. N/N+1	Program unit No.	Program unit No.

The program unit for the odd device numbers is on bits 1 to 9.

That for the even device numbers is on bits 10 to 18.

11.1 Relationship between the "Device Number to Program Unit Number" Table and the "Program Unit Number to Device Control Field" Table

See the description of the "Program Unit to Device Control Field" table.

For notes

12 Description of the SC Memory Cell Assignment in System 8870/4

Symbol	Address	Meaning
UINT1A	0.0. 0	Contains accumulator content when there is an interruption through interrupt level 1 (call address of the I/O interrupt end routine).
XI	0.0. 2	Index register for indexing of micro-instructions.
XAHW YINT2A	0.0. 4	Accumulator content when there is an interruption through level 2.
YINT1U	0.0. 6	Store for 901 micro-instruction addresses when there is an interruption through interrupt level 1.
XBZHW YINT2U	0.0. 8	As above, interrupt level 2.
YMIN	0.0. A	Contains constants 3, 15, 15, 15, 15, (required for address modifications).
YESART	0.0. C	Flag memory for switching-on modes: 3.1.0.2.0 System initialisation 3.2.0.0.4 Sermik from disk 3.3.4.0.0 SYSTOOL call = (IPL) 0.0.0.0.2 no memory test 0.0.0.0.1 do not initial load
XFE	0.0. E	Flag memory for error or parity interrupt: 0.0.0.0.2 Error interrupt 0.0.0.0.1 Parity interrupt 0.0.0.0.0 Power failure interrupt
XDIAU	0.1. 0	Back space address memory for diagnosis routine sub-program.
XINIU	0.1. 2	Back space address memory for sub-program on initialisation call.
XDIA1	0.1. 4	Auxiliary cell 1 Used in initial-
XDIA2	0.1. 6	2 isation phase,
XDIA3	0.1. 8	3 loading process
XDIA4	0.1. A	4 and memory test.
XEPZG	0.1. C	Address indicator for the error and parity interrupt error handling routine call address.
XNAZG	0.1. E	Address indicator for power failure interrupt error handling routine call address.

12.1 Initial Program Loader Working Area in 8870/4

The initial loader working area is only required for initialisation and restart after power failure. It occupies the upper area of the I/O stack pointer, that is to say, the area from address 0.B.C. to 1.F.E.

Since this area must always be kept free for the loader, no interrupt number greater than 5.E = 94 may be used in system 8870/4.

Assignment of initial loader area:

Address	Meaning
0.B.C 0.D.0	Loader working cells
0.D.2 1.F.E	Buffer area to record the instructions

12.2 Assignment of the Global Cells

Symbol	Address	Meaning
Y Bit 13	2.0.0	Auxiliary global cells for high bits 13 to 18.
Y Bit 14	2.0.2	
Y Bit 15	2.0.4	These bits are used to build block
Y Bit 16	2.0.6	addresses or to amend label addresses
Y Bit 17	2.0.8	with block addresses.
Y Bit 18	2.0.A	
YSIMZ	2.0.C	These cells contain the "Change from Simultaneous to Central Running" program routine call address.
YZSIM	2.0.E	Contains the call address of the "Perform System Function" sub-program. System functions, time management, short time area management, etc.
YJBOV	2.1.2	Contains the call address of the "Job Scheduling" sub-program.

Symbol	Address	Meaning
YINTNU	2.1.4	Contains the entry address of the 2nd level interrupt routine.
YZABUP	2.1.6	Contains the call address of the "Central Load Shifting and Signing-Off Sub-Program".
YEZEIT	2.1.8	Contains the present time: Bits 1 to 6 = seconds from 0 to 59 Bits 7 to 12 = minutes from 0 to 59 Bits 13 to 18 = hours from 0 to 23
YWECK	2.1.A	Contains the call address of the "Bell Activation" sub-program.
YWKANF	2.1.C	Contains the start address of the active bell in the bell table.
YWKEND	2.1.E	Contains the end address of the active bells in the bell table.
YVVAK1	2.2.0	Variable connector 1, Job scheduler cell contains either the call address of the routine for job field request or the call address of the "Process Job Scheduler Queues" routine.
XVVAK2	2.2.2	Variable connector 2, Job scheduler cell required when job fields are connected (meaning as above).
XWSIM2	2.2.4	Auxiliary bell cell: contains the start address of the bell which should be the next to be processed.
XUZG1	2.2.6	Contains the call address of the clock interrupt routine. With clock interrupt jump to (XUZG1).
XUTAKT	2.2.8	Contains the value 3.15.10.7.1 for the CPU clock 4 ms cycle.
YRESRV	2.2.A	Contains the call address of the clock interrupt end routine (UHRWR) or zero. Used for TACOS.
YOSAD	2.2.C	Cell to distinguish between OSA and OSD. VISAD = 0.0.0.0.2 = OSD.
YDOUBL	2.2.E	States whether the machine is operating singly or doubly. YDOUBL = 0 = Fore- or single operation YDOUBL = 1 = Background.

Symbol	Address	Meaning
YFMIN	2.3.0	Carry collector for foreground 15.8 min.
YFNKS	2.3.2	4 ms counter for foreground. The counter continues to operate after 15.8 min. The carry is stored in YFMIN.
YBMIN	2.3.4	Carry collector for background is 15.8 min.
YBMLS	2.3.6	4 ms counter for background. The counter continues to operate after 15.8 min. The carry is stored in YBMIN.
YDSZHZ	2.3.8	Contains the DMA nos. which are at present occupied by the system. Only used as auxiliary cell.
YFEADR	2.3.A	Contains the address of the micro-instruction where a parity or time error can be recognized.
YUHILF	2.3.C	Frequency reducer 250 : 1. Required to ascertain the ceconds. 250x4ms = 1 s.
YSEK	2.3.E	Second counter (i.e. counter of seconds).
YISNU	2.4.0	Contains the present short time index if the system is operating simultaneously or zero if it is operating centrally.
YSYPAR	2.4.2	Contains the start address of the system output table.
XVKOTB	2.4.4	Contains the start address of the device combination table.
YGKFTB	2.4.6	Contains the start address of the "Program unit to Device Control Field" table.
YTNLTB XU8	2.4.8	Contains the start address of the "Program unit to Long-Time Area" table. Used in the Sermik as power failure indicator flag.
YETB XU10	2.4.A	Contains the negated end address of the "Program unit to Long-Time Area" table.
YGERTN	2.4.C	Contains the start address of the "Device Number to Program Unit Number" table.
YGTEND	2.4.E	Contains the negated end address of the "Device Number to Program Unit Number" table.
YINKTB	2.5.0	Contains the start address of the "Interrupt Number to Short time Area" table.

Symbol	Address	Meaning
YWTAB	2.4.2	Contains the start address of the "Bell to Short time Area" table.
YWNR	2.5.4	Contains the largest bell number.
YHSSAS	2.5.6	Contains the start address of the main memory.
YHSDAS	2.5.8	Contains the address of the first free byte after the device control fields.
YHSEND	2.5.A	Contains the end address of the lower main memory.
YJFF1	2.5.C	Contains the address of a job field in the job field chain which is still free, or zero if no more free job fields are available in the chain.
YKZK1	2.5.E	Contains the address of a Class 1 short time area that is still free, or zero if no further short time areas of this class are available.
YKZK2	2.6.0	Contains the address of a class 2 short time area that is still free, or zero if no further areas of this class are available.
YKZK3	2.6.2	Means the same as YKZK1, but for class 3
YKZK4	2.6.4	Means the same as YKZK1, but for class 4
YKZK5	2.6.6	Means the same as YKZK1, but for class 5
YKZK6	2.6.8	Means the same as YKZK1, but for class 6
YKZK7	2.6.A	Means the same as YKZK1, but for class 7
YKZK8	2.6.C	Means the same as YKZK1, but for class 8
YKZK9	2.6.E	Menas the same as YKZK1, but for class 9
YKZK10	2.7.0	Means the same as YKZK1, but for class10
YKZK11	2.7.2	Means the same as YKZK1, but for class11
YKZK12	2.7.4	Means the same as YKZK1, but for class12
YKZK13	2.7.6	Means the same as YKZK1, but for class13
YKZK14	2.7.8	Means the same as YKZK1, but for class14
YKZK15	2.7.A	Means the same as YKZK1, but for class15
XVUSY YAR1	2.7.C	Either contains the rewind address of the program routine which has called a system function, or is used as a working cell during simultaneous work.

Symbol	Address	Meaning
XVUJO YAR2	2.7.E	Either contains the return address of the program routine which schedules a job, or is used as a working cell during simultaneous working.
XNETZA	2.8.0	Contains the accumulator content in the event of recognised power failure.
XNETZU	2.8.2	Contains the address of the micro-instruction where the power failure interrupt can be recognised.
XNETZI	2.8.4	Contains the index register content which was current when power failure was recognised.
XFHZ1	2.8.6	Three auxiliary cells for processing parity or time error interrupts.
XFHZ2	2.8.8	
XFHZ3	2.8.A	
XAW	2.8.C	When bell runs out, contains the accumulator content which was current when the clock interrupt was recognised.
XIW	2.8.E	When bell runs out, contains the index register content which was current when the clock interrupt was recognised.
XBZW	2.9.0	When bell runs out, contains the address of the micro-instruction which was current when the clock interrupt was recognised.
XVBZSW	2.9.2	During job scheduling, contains the instruction address of the program interrupted by clock or I/O interrupt.
XVISW	2.9.4	During job scheduling, contains the index value of the program interrupted by clock or I/O interrupt.
XVASW	2.9.6	During job scheduling, contains the accumulator content of the program interrupted by clock or I/O interrupt.
XVEIN	2.9.8	Job scheduler entry counter. Contains the quantity of jobs to be connected.
XIHW	2.9.A	During interrupt performance, contains the index value of the program interrupted by an I/O interrupt.

Symbol	Address	Meaning
XI1	2.9.C	Contains either the table place address of the "Interrupt no. to short time area" table, when I/O interrupt processing is pending, or the index of the program which has called in the job scheduler (at time of job scheduling).
XVAEW1	2.9.E	Start pointer of job scheduler input queue 1.
XVEEW1	2.A.0	End pointer of same.
XVAAW1	2.A.2	Start pointer of job scheduler output queue 1.
XVEAW1	2.A.4	End pointer of same.
XVAEW3	2.A.6	Start pointer of job scheduler input queue 3.
XVEEW3	2.A.8	End pointer of same.
XVAAW3	2.A.A.	Start pointer of job scheduler output queue 3.
XVEAW3	2.A.C.	End pointer of same.
XUWMK	2.A.E	Contains the return address of the program interrupted by clock interrupt while the clock interrupt is being dealt with. Contains zero if the system is operating centrally.
XVFEB	2.B.0	Contains error flags if an error has been discovered in the job scheduler, e.g. absence of operating media.
XVJSTA	2.B.2	Contains the status of the job scheduler. (XVJSTA) = 0, job scheduler status passive. (XVJSTA) ≠ 0, job scheduler active. In the initialisation phase, XVJSTA is used as XMVB. In this case it contains the start address of the module supply blocks.
XVAKW	2.B.4	Pointer for present output queue. Contains the address of the present output queue pointer.
XVDSZ	2.B.6	Contains the number of the DMA currently assigned.

Symbol	Address	Meaning
XVZA	2.B.8	Start pointer of the queue for time request.
XVZE	2.B.A	End pointer of same.
XVSUZG	2.B.C	Device combination table pointer.
XVBAKT	2.B.E	Contains the device combination bits. Not exploited at present.
XVJEA	2.C.0	Start pointer of the queue for job field request.
XVJFE	2.C.2	End pointer of same.
XVH1	2.C.4	Job scheduler auxiliary cell 1.
.	.	.
.	.	.
.	.	.
.	.	.
XVH11	2.D.8	Job scheduler auxiliary cell 11.
XVH12	2.D.A	Initialisation auxiliary cell.
.	.	.
.	.	.
.	.	.
.	.	.
XVH14	2.D.E	Initialisation auxiliary cell.

13 Description of the First 256 Working Cells in 8870/6 (These are located in CPU 1501)

Symbol	Address CPU	Address 901	Meaning
XINT1A	0.0	0.0. 0	Contains the accumulator content at the time of interruption at interrupt level 1.
XI	0.1	0.0. 2	Index register for indexing of micro-instructions 901.
XAHW YINT2A	0.2	0.0. 4	Accumulator content when there is an interruption through interrupt level 2.
YINT1U	0.3	0.0. 6	Store for 901 micro-instruction addresses when there is an interruption through interrupt level 1.
YBZHW YINT2U	0.4	0.0. 8	As above, interrupt level 2.
YMIN	0.5	0.0. A	Contains constants 3.15.15.15.15 (required for ADR modifications).
YESART	0.6	0.0. C	Flag memory for switching on modes: 3.1.0.2.0 System initialisation 3.2.0.0.4 Sermik from disk 3.3.4.0.0 SYSTOOL call = (IPL) 0.0.0.0.2 no memory test 0.0.0.0.1 do not initial load
XFE	0.7	0.0. E	Flag memory for error or parity interrupt: 0.0.0.0.2 Error interrupt 0.0.0.0.1 Parity interrupt 0.0.0.0.0 Power failure interrupt
XDIAU	0.8	0.1. 0	Return address memory for diagnosis routine sub-program.
XINIU	0.9	0.1. 2	Return address memory for sub-program on initialisation call.
XDIA1 XDIA2 XDIA3 XDIA4	0.A 0.B 0.C 0.D	0.1. 4 0.1. 6 0.1. 8 0.1. A	Auxiliary diagnosis cell 1, 2, 3, 4 } Used in IPL loading & memory test
XEPZG	0.E	0.1. C	Address pointer for the error and parity interrupt error handling routine call address.
XNAZG	0.F	0.1. E	Address pointer for power failure interrupt error handling routine call address.

Symbol	Address CPU	Address 901	Meaning
SATZTAB	1.0	0.2. 0	contains the start address of the foreground register lable of the macro-interpreter or of the TPC, if TACOS is operating.
XIMA01	1.1	0.2. 2	Contains the start address of the background register label.
XIMA02	1.2	0.2. 4	Start address of TVP1 foreground register label, when TACOS is operating.
XIMA03	1.3	0.2. 6	Start address of TVP2 foreground register label, when TACOS is operating.
XIMA04	1.4	0.2. 8	Start address of TVP3 foreground register lable, when TACOS is operating.
XIMA05	1.5	0.2. A	Start address of TVP4 foreground register lable, when TACOS is operating.
XIMA06	1.6	0.2. C	Start address of TVP5 foreground register lable, when TACOS is operating.
XIMA07	1.7	0.2. E	Start address of TVP6 foreground register lable, when TACOS is operating.
XIMA08	1.8	0.3. 0	Start address of TVP7 foreground register lable, when TACOS is operating.
XIMA09	1.9	0.3. 2	Start address of TVP8 foreground register lable, when TACOS is operating.
XIMA0A	1.A	0.3. 4	Start address of TVP9 foreground register lable, when TACOS is operating.
XIMA0B	1.B	0.3. 6	Start address of TVP10 foreground register lable, when TACOS is operating.
XIMA0C	1.C	0.3. 8	Start address of TCP11 foreground register lable, when TACOS is operating.
XIMA0D	1.D	0.3. A	Start address of TVP12 foreground register lable, when TACOS is operating.
XIMA0E	1.E	0.3. C	Start address of TVP13 foreground register lable, when TACOS is operating.

Symbol	Address CPU	Address 901	Meaning
XIMA0F	1.F	0.3. E	Start address of TVP14 foreground register lable, when TACOS is operating.
XIHW	2.0	0.4. 0	Buffer for the present index value of the program interrupt by a I/O interrupt.
YISNU	2.1	0.4. 2	Contains present short time index if the system is operating simultaneously, or zero when it is operating centrally.
YINTB	2.2	0.4. 4	Contains the start address of the "Interrupt no. to short time area" table.
YSIEND	2.3	0.4. 6	Memory for simultaneous flags: 1 = bell call 0 = I/O interrupt call
XMAXIN	2.4	0.4. 8	Contains the largest I/O interrupt no. which occurs in the system on bits 8 to 18.
XMOVE1	2.5	0.4. A	Auxiliary cell for micro-move. The content of XBZ = SR1 is transferred to XMOVE1.
XMOVE2	2.6	0.4. C	Auxiliary cell for micro-move. The content of XQL = SR6 is transferred to XMOVE2.
XMOVE3	2.7	0.4. E	Auxiliary cell for micro-move. The content of XZA = SR7 is transferred to XMOVE3.
XI11	2.8	0.5. 0	Contains the address of the place in the "SATZTAB" table of the micro-program at present active.
XITVP	2.9	0.5. 2	Contains the address of the place in the "SATZTAB" table of the next TVP in the foreground to be called.
			Global cells for following constants:
YBIT13	2.A	0.5. 4	0.1.0.0.0
YBIT14	2.B	0.5. 6	0.2.0.0.0
YBIT15	2.C	0.5. 8	0.4.0.0.0
YBIT16	2.D	0.5. A	0.8.0.0.0
YBIT17	2.E	0.5. C	1.0.0.0.0
YBIT18	2.F	0.5. E	2.0.0.0.0

Symbol	Address CPU	Address 901	Meaning
YSIMZ	3.0	0.6. 0	Contains the start address of the program routine for changing from simultaneous to central.
YESIM	3.1	0.6.2	Contains the start address of the program routine for changing from central to simultaneous.
YSYSV	3.2	0.6.4	Contains the call address of the "Perform System Functions" sub-program.
YJOBV	3.3	0.6. 6	Contains the call address of the sub-program for job scheduling.
YZABUP	3.4	0.6. 8	Contains the call address of the central contacting and signing off sub-program.
YWECK	3.5	0.6. A	Contains the call address of the sub-program for bell processing.
XVVAK1	3.6	0.6. C	Address pointer 1 for the variable connectors. Contains either the call address for job scheduler or that for the job management routine.
XVVAK2	3.7	0.6. E	Address pointer 2 for the variable connectors. Assigned as in the case of address pointer 1.
YINTNU	3.8	0.7. 0	Contains the call address of the program routine for interrupt end processing.
XWSIM2	3.9	0.7. 2	Auxiliary cell for bell simulation.
YXIMZ1	3.A	0.7. 4	Contains the call address of the program routine for changing from simultaneous to central, with storage of the return address.
YWERT10	3.B	0.7. 6	Contains the constants 0.0.0.0.10.
XAW	3.C	0.7. 8	Buffer for the accumulator content of the program interrupted by clock-interrupt. (XAHW) ⟶ XAW.
XIW	3.D	0.7. A	Buffer for the index value of the program interrupted by clock-interrupt. (XI) ⟶ XIW.
XVBZ	3.E	0.7. C	Buffer for the instruction address of the program interrupted by clock-interrupt. (XINT2U) ⟶ XBZW.

Symbol	Address CPU	Address 901	Meaning
XVBZSW	3.F	0.7. E	Buffer for the instruction address in the job scheduler routine. The instruction address refers to the program interrupted by clock or I/O interrupt before job scheduling is called.
XVISW	4.0	0.8. 0	Buffer for the index value of the program interrupted by clock or I/O interrupt before job scheduling routine is called.
XVASW	4.1	0.8. 2	Buffer for the accumulator content of the program interrupted by clock or I/O interrupt before job scheduler routine is called.
XVEIN	4.2	0.8. 4	Job scheduler entry counter. Contains the quantity of jobs to be connected.
XI1	4.3	0.8. 6	Buffer for the start address of the short time area of the jobs to be connected. Required in the job scheduler routine.
XVAEW1	4.4	0.8. 8	Start of job scheduler input queue 1.
XVEEW1	4.5	0.0. 8	End of same.
XVAAW1	4.6	0.8. C	Start of job scheduler output queue 1.
XVEAW1	4.7	0.8. E	End of same.
XVAEW3	4.8	0.9. 0	Start of job scheduler input queue 3.
XVEEW3	4.9	0.9. 2	End of same.
XVAAW3	4.A	0.9. 4	Start of job scheduler output queue 3.
XVEAW3	4.B	0.9. 6	End of same.
XUWMK	4.C	0.9. 8	Flag for the clock interrupt routine. Contains the instruction address of the program interrupted by clock interrupt. Contains zero when the system is operating centrally.
YEANR	4.D	0.9. A	Memory for the I/O interrupt no. which was processed last.
XMVB XVISTA	4.E	0.9. C	Start address of the module supply block of the present program unit or memory for the job scheduler status.

Symbol	Address CPU	Address 901	Meaning
XMODNR XVAKAW	4.F	0.9. E	Contains the present module number or is used as a pointer for the present job scheduler output queue.
XH1	5.0	0.A. 0	Working cell for the macro-interpreter.
XU1	5.1	0.A. 2	Working cell for the macro-interpreter. Used to store the return address in the case of stage 1 sub-programs.
XQR	5.2	0.A. 4	Working cell for the macro-interpreter. Used to store source field addresses.
XOP	5.3	0.A. 6	Working cell for the macro-interpreter. Used to store the device number or micro-instruction OP-codes.
XI3	5.4	0.A. 8	Macro-interpreter working cell. Used to record the start address of the long-time area of a program unit which is to be processed.
XHGK	5.5	0.A. A	Macro-interpreter working cell. Used as an auxiliary cell.
XUPUP	5.6	0.A. C	Macro-interpreter working cell. Used as an auxiliary cell.
XU2	5.7	0.A. E	Macro-interpreter working cell. Used to store the return address in the case of stage 2 sub-programs.
XHA	5.8	0.B. 0	Macro-interpreter working cell. Used as an auxiliary cell.
XU3	5.9	0.B. 2	Macro-interpreter working cell. Used to store the return address in the case of stage 3 sub-programs.
YTEST1 YTEST2 YTEST3	5.A 5.B 5.C	0.B. 4 0.B. 6 0.B. 8	Communications cell between item being tested and test machine.
XH14	5.D	0.B. A	Macro-interpreter working cell. Used as an auxiliary cell.
XH15	5.E	0.B. C	Macro-interpreter working cell. Used as an auxiliary cell.
XH16	5.F	0.B. F	Macro-interpreter working cell. Used as an auxiliary cell.

Symbol	Address CPU	Address 901	Meaning
XABO	6.0	0.C.0	Working cell for the macro-interpreter. Contains zero if the call phase is to be called; otherwise contains the call address of the MATABO program routine (to check out whether a program change should be effected).
XAD	6.1	0.C.2	Working cell for the macro-interpreter. Contains data on the history of address generation (if bits 12 = 1, register address characters; if bit 1 = 1, dual address characters) or the status of the optional CPU flags when the macro-interpreter is called.
XIR3	6.2	0.C.4	Working cell for the CPU micro. Contains return address for interrupt processing.
YEZEIT	6.3	0.C.6	Memory for real time.
YAR1 XVUSY	6.4	0.C.8	Global simultaneous working cell. Used on the occasion of job scheduling and interrupt processing.
YAR2 XVUJO	6.5	0.C.A	Global simultaneous working cell. Used on the occasion of job scheduling and interrupt processing.
YDOUBL	6.6	0.C.C	Cell for program allocation: 0 = foreground 1 = background
YFMIN	6.7	0.C.E	Foreground time counter for CPU time calculation. Counter for the time that the 4 ms counter exceeds 15.8 mins.
YFMLS	6.8	0.D.0	Foreground time counter for CPU time calculation. Goes up 1 every 4 ms. Max. reading 15.8 mins. The excess is counted in YFMIN.
YBMIN	6.9	0.D.2	Background time counter for CPU time calculation. Meaning as YFMIN.
YBMLS	6.A	0.D.4	Background time counter for CPU time calculation. Meaning as YFMLS.
YUHILF	6.B	0.D.6	Frequency reducer 250 : 1. Required to ascertain the seconds. 250 x 4 ms = 1 s.

Symbol	Address CPU	Address 901	Meaning
YSEK	6.C	0.D.8	Second counter (i.e. counter of seconds).
XPWD	6.D	0.D.A	Flag cell for wish change.
XHSSAS	6.E	0.D.C	Contains macro-memory start address
XFORE	6.F	0.D.E	Contains the quantity of peripheral devices operating in the fore-ground.
XBZ	7.0	0.E.0	Macro-instruction address pointer. Always points to the macro-instruction byte which is at present being processed.
XLF	7.1	0.E.2	Contains the current N-address of the macro-instruction being processed.
XVK	7.2	0.E.4	Used as a buffer for the macro-instruction pre-code character.
XAR	7.3	0.E.6	Used as an auxiliary cell for macro-instruction processing.
XZAN	7.4	0.E.8	Used as an auxiliary cell for macro-instruction processing.
XZI	7.5	0.E.A	Contains the full N-address 1 of the macro-instruction at present being processed.
XQL	7.6	0.E.C	Contains the full M-address of the macro-instruction at present being processed.
XZA	7.7	0.E.E	States the length of the M-field for VS, V1, VF, VSL, V1L and VFL pre-code character instructions.
XZAM	7.8	0.F.0	Contains the code of the end character for VU, VUL, VA and VAL pre-code character instructions.
YRINT	7.9	0.F.2	CPU micro working cell. Contains the present flag status for the CPU internal interrupt inhibit register.
YRRU	7.A	0.F.4	CPU micro working cell. Contains the address of whichever jump instruction was last carried out in the 901 micro.
YRAKKU	7.B	0.F.6	CPU micro working cell. Contains the outcome of this 901 micro-instruction (= accumulator content).

Symbol	Address CPU	Address 901	Meaning
YRBZMI	7.C	0.F.8	Working cell for the CPU micro. Instruction counter for micro 901.
YRCEFE	7.D	0.F.A	Working cell for the CPU micro. Contains the constant 0.0.15.15.14. Required when processing 901 micro-instruction in order to be able to mask the OP part.
YRCFFF	7.E	0.F.C	Working cell for the CPU micro. Contains the constant 0.0.15.15.15. Required when processing 901 micro-instructions in order to be able to mask the OP part.
YRINT	7.F	0.F.E	Working cell for the CPU micro. Contains the constant 0.2.0.1.5. Required in order to be able to insert the status register when defined.

For notes

14 Description of the System and Index Registers

14.1 System Registers

Nr.	Symbol	Address CPU	Address 901	Meaning
0	X0S	8. 0	1.0. 0	System registers X05 to X315 are
1	X1S	8. 1	1.0. 2	with the corresponding index
2	X2S	8. 2	1.0. 4	registers each time there is a
3	X3S	8. 3	1.0. 6	change of level.
4	XBPS	8. 4	1.0. 8	
5	XDAS	8. 5	1.0. A	Contains the start address of the
6	XMAS	8. 6	1.0. C	system working area.
7	XICS	8. 7	1.0. E	
8	XMKS	8. 8	1.1. 0	
9	XFLS	8. 9	1.1. 2	
10	XRAS	8. A	1.1. 4	
11	XLCS	8. B	1.1. 6	
12	XSCS	8. C	1.1. 8	
13	XIOCSS	8. D	1.1. A	
14	XB1S	8. E	1.1. C	
15	XB2S	8. F	1.1. E	
16	X16S	9. 0	1.2. 0	
17	X16S	9. 1	1.2. 2	
18	XBP1S	9. 2	1.2. 4	
19	XIC1S	9. 3	1.2. 6	
20	XBP2S	9. 4	1.2. 8	
21	XIC2S	9. 5	1.2. A	
22	XBP3S	9. 6	1.2. C	
23	XIC3S	9. 7	1.2. E	
24	XBP4S	9. 8	1.3. 0	
25	XIC4S	9. 9	1.3. 2	
26	XBP5S	9. A	1.3. 4	
27	XICS5	9. B	1.3. 6	
28	XBP6S	9. C	1.3. 8	
29	XIC6S	9. D	1.3. A	

No.	Symbol	Address CPU	Address 901	Meaning
30 31	XBP7S XIC7S	9. E 9. F	1.3. A 1.3. E	Always contains the present program reference number and the sub-program return address.
32	XSAS	A. 0	1.4. 0	Start address of the byte memory.
33	XSLB	A. 1	1.4. 2	Start address of the free user area.
34	XLAD	A. 2	1.4. 4	Loading address for the user program.
35	XEND	A. 3	1.4. 6	Contains the address of the loading instruction which occupies the last 7 bytes of the core memory. XEND + 6 = XSUB
36	XPW	A. 4	1.4. 8	Contains the program interrupt request bits.
37	XSW	A. 5	1.4. A	Contains the program status: 0 = system level 3.15.15.15.15 = user level
38	XSUPA	A. 6	1.4. C	Contains the address part in the case of PRG instruction.
39	XSUK	A. 7	1.4. E	Contains the subsequent character in the case of PRG instruction.
40	XCOM	A. 8	1.5. 0	Contains the start address of the common area.
41	XDIBU	A. 9	1.5. 2	Contains the start address of the file assignment list with associated disc buffer.
42	XOVL	A. A	1.5. 4	Contains the start address of the overlay area.
43	XTST 1	A. B	1.5. 6	Contains the register number for the test stop.
44	XTST 2	A. C	1.5. 8	Contains the comparative data for XSTS1.
45	XTXT 3	A. D	1.5. A	Contains the test stop address.
46	XFJOB	A. E	1.5. C	Quantity of devices operating in the foreground (excluding and line).
47	XOSA	A. F	1.5. E	Contains the OSD call address.
48	XIO 1	D. 0	1.6. 0	Contains the start address for the IOCS routines GET and PUT.
49	XIO 2	D. 1	1.6. 2	Contains the start address for the IOCS routines OPEN and CLOSE.

50 to 56: these system registers are assigned and managed by the loading program.

No.	Symbol	Address CPU	Address 901	Meaning
50	XD	B. 2	1.6. 4	Contains the address of the working and I/O area for the first phase of the user program.
51	XDH	B. 3	1.6. 6	Contains the address of the working and I/O area for the main phase just loaded.
52	XDB	B. 4	1.6. 8	Contains the address of the working and I/O area for the secondary phase just loaded.
53	XLH	B. 5	1.6. A	Contains the loading address of the main phase just loaded.
54	XLB	B. 6	1.6. C	Contains the loading address of the secondary phase just loaded.
55	XCSH	B. 7	1.6. E	Contains the start address of the loaded constant segment for the main phase.
56	XCSB	B. 8	1.7. 0	Contains the start address of the loaded constant segment for the secondary phase.
57	XTST4	B. 9	1.7. 2	Auxiliary register for the test system.
58	XTST5	B. A	1.7. 4	Auxiliary register for the test system.
59	XSUB	B. B	1.7. 6	Contains the start address of the last byte in the system transfer area - end of the core memory.
60	XDOUB1	B. C	1.7. 8	Contains the start address of the entry point for various system functions in double.
61	XDOUB2	B. D	1.7. A	Auxiliary register for double module.
62	XDISK	B. E	1.7. C	Contains the call address of the IOCS board.
63	XDAWF	B. F	1.7. E	XDAWF = XDA = Start address of the data area.

14.2 Index Registers

No.	Symbol	Address CPU	Address 901	Meaning
0	X0	C.0	1.8.0	Contains the start address of the last object field of data field instructions.
1	X1	C.1	1.8.2	Contains the start address of the next object field for data field instructions.
2	X2	C.2	1.8.4	Contains the start address of the last source field of data field instructions.
3	X3	C.3	1.8.6	Contains the start address of the next source field of data field instructions.
4	XBP	C.4	1.8.8	Contains the currently valid program reference point address (= start address of the corresponding program routine).
5	XDA	C.5	1.8.A	Contains the start address of the data area.
6	XMA	C.6	1.8.C	Contains the start address of the mask area.
7	XIC	C.7	1.8.E	Contains the start address of the current macro-instruction.
8	XMK	C.8	1.9.0	Flag register, flag assignment from bit 1 to 16.
9	XFL	C.9	1.9.2	Contains the length of the sliding point field for sliding point arithmetic.
10	XRA	C.A	1.9.4	Contains the address of the loop start for repeat loop instruction.
11	XLC	C.B	1.9.6	Contains the number of times the loop is to be run through in the case of repeat loop or loop instructions.
12	XSC	C.C	1.9.8	Sub-program stage counter = Program stage number.
13	XIOCS	C.D	1.9.A	Work register for IOCS.
14	XB1	C.E	1.9.C	Free for the user (except in the case of TACOS).
15	XB2	C.F	1.9.E	
16	X16	D.0	1.A.0	
17	X17	D.1	1.A.2	
18	XBP1	D.2	1.A.4	Contains the start address of program stage 0 (= main program when sub-program stage 1 is called in).

Nr.	Symbol	Address CPU	Address 901	Meaning
19	XIC1	D.3	1.A.6	Contains the return address into program stage 0 when calling subprogram stage 1.
20	XBP2	D.4	1.A.8	Contains the start address of subprogram stage 1 when calling subprogram stage 2.
21	XIC2	D.5	1.A.A	Contains the return address into program stage 1 when calling subprogram stage 2.
22	XBP3	D.6	1.A.C	Contains the start address of subprogram stage 2 when calling subprogram stage 3.
23	XIC3	D.7	1.A.E	Contains the return address into program stage 2 when calling subprogram stage 3.
24	XBP4	D.8	1.B.0	Contains the start address of subprogram stage 3 when calling subprogram stage 4.
25	XIC4	D.9	1.B.2	Contains the return address into program stage 3 when calling subprogram stage 4.
26	XBP5	D.A	1.B.4	Contains the start address of subprogram stage 4 when calling subprogram stage 5.
27	XIC5	D.B	1.B.6	Contains the return address into program stage 4 when calling subprogram stage 5.
28	XBP6	D.C	1.B.8	Contains the start address of subprogram stage 5 when calling subprogram stage 6.
29	XIC6	D.D	1.B.A	Contains the return address into program stage 5 when calling subprogram stage 6.
30	XBP7	D.E	1.B.C	Contains the start address of subprogram stage 6 when calling subprogram stage 7.
31	XIC7	D.F	1.B.E	Contains the return address into program stage 6 when calling subprogram stage 7.
32	X32	E.0	1.C.0	These registers remain freely available to the user.
.	.	.	.	
.	.	.	.	
.	.	.	.	
63	X63	F.F	1.C.E	

Symbol	Address	Meaning
YWKANF	2.0.0	Start of the bell chain.
YWKEND	2.0.2	End of the bell chain.
XUTAKT	2.0.4	Cell for time cycle.
YDSZHZ	2.0.6	Auxiliary cell for DMA assignment.
YFEADR	2.0.8	Error address of the last parity or error interrupt.
SYSPAR	2.0.A	Start of the system output table.
XCKOTB	2.0.C	Start of the device combination table.
YGKFTB	2.0.E	Start of the "Program unit to Device Control Field" table.
YTNLTB	2.1.0	Start of the "Program unit to Long-Time Area" table.
YETB	2.1.2	Negated end address of same.
YGERTN	2.1.4	Start of the "Device Number to Program Unit" table.
YGTEND	2.1.6	Negated end address of same.
YWTAB	2.1.8	Start of the "Bell to Short time Work Area" table.
YWNR	2.1.A	Highest bell number.
XINTER	2.1.C	Internal failure switch.
YHSDAS	2.1.E	First free byte after the device control fields.
YHSEND	2.2.0	End of main storage (last byte).
XFHZ1	2.2.2	Auxiliary cell for processing parity or error interrupt.
XFHZ2	2.2.4	Auxiliary cell for processing parity or error interrupt.
XUZG1	2.2.6	Time indicator (switch in clock interrupt).
UJFF1	2.2.8	Indicator of first free job field.
YKZK1	2.2.A	Short time class pointer 1.
YKZK2	2.2.C	Short time class pointer 2.
YKZK3	2.2.E	Short time class pointer 3.
YKZK4	2.3.0	Short time class pointer 4.
YKZK5	2.3.2	Short time class pointer 5.
YKZK6	2.3.4	Short time class pointer 6.
YKZK7	2.3.6	Short time class pointer 7.
YKZK8	2.3.8	Short time class pointer 8.

Symbol	Address	Meaning
YKZK9	2.3.A	Short time class pointer 9.
YKZK10	2.3.C	Short time class pointer 10.
YKZK11	2.3.E	Short time class pointer 11.
YKZK12	2.4.0	Short time class pointer 12.
YKZK13	2.4.2	Short time class pointer 13.
YKZK14	2.4.4	Short time class pointer 14.
YKZK15	2.4.6	Short time class pointer 15.
XU8	2.4.8	Sermik flag/"N/A" indicator (N/A = power failure).
XU10	2.4.A	Sermik flag/"N/A" indicator (N/A = power failure).
YGZUHR	2.4.C	TONS global cell.
XIAR	2.4.E	3 auxiliary cells for initialisation.
XVDSZ	XIAR	Present DMA assignment.
XIAR2	2.5.0	Start of the queue for time request.
XVZA	XIAR2	
XIAR3	2.5.2	End of same.
XVZE	XIAR3	
XUI1	2.5.4	UP store 1 initialisation.
XVSUZG	XIU1	Device Combination Table indicator.
XIU2	2.5.6	UP store 2 initialisation.
XVBAKT	XIU2	Present device combination.
XAPE2	2.5.8	Word 2 of an UP lable.
XVJFA	XAPW2	Start of queue for job field request.
XIZAEL	2.5.A	Auxiliary cell 2 initialisation.
XVJFE	XIZAEL	End of queue for job field request.
XAP	2.5.C	Installation parameter pointer.
XVH1	XAP	Job scheduler auxiliary cell.
XHW	2.5.E	Auxiliary cell for initialisation.
XVH2	XHW	Job scheduler auxiliary cell.
XBITZ	2.6.0	Auxiliary cell for determining the quantity of sequence parameters.
XVH3	XBITZ	Job scheduler auxiliary cell.
XINIF	2.6.2	Error on initialisation.
XVH4	XINIF	Job scheduler auxiliary cell.

Symbol	Address	Meaning
XTNNR	2.6.4	Terminal number.
XVH5	XTNNR	Job scheduler auxiliary cell.
XUNTEN	2.6.6	Auxiliary cell: "engage area below".
XINTNR	XUNTEN	Interrupt number.
XVH6	XUNTEN	Job scheduler auxiliary cell.
XAP2	2.6.8	Installation parameter reserve pointer.
XVH7	XAP2	Job scheduler auxiliary cell.
XAPR	2.6.A	UP reserve indicator.
XVH8	XAPR	Job scheduler auxiliary cell.
XGERNR	2.6.C	Device number.
XVH9	XGERNR	Job scheduler auxiliary cell.
XLZ	2.6.E	Mark indicator.
XVH10	XLZ	Job scheduler auxiliary cell.
XMVB2	2.7.0	Module start address.
XVH11	XMVB2	Job scheduler auxiliary cell.
XIAR4	2.7.2	Initialisation working cell.
XVH12	XIAR4	Job scheduler auxiliary cell.
XIU3	2.7.4	Initialisation UP store.
XVH13	XIU3	Job scheduler auxiliary cell.
XIU4	2.7.6	Initialisation UP store.
XVH14	XIU4	Job scheduler auxiliary cell.
XINTF	2.7.8	Internal error number.
XPRG	2.7.A	Start address of the operating system.
XMAGK	2.7.C	Start address of the GK 904 micro or zero.
XMATST	2.7.E	Start address of the test module, or zero.
XVFEB	2.8.0	Flag for "absence of operating media".
XTONS	2.8.2	Start address of TONS-UP or zero.
YSAB	2.8.4	Start address of SYSARB-1 to be transferred
YPRUEF	2.8.6	Flag cell for CPU error (for FE); in the event of failure (YPRUEF) = 2.0.0.
XITONS	2.8.8	TONS index global cell.
XFHZ3	2.8.A	Auxiliary cell for processing parity or error interrupt.

Symbol	Address	Meaning
XQ1	2.8.C	Global cell for IOCS line.
XQ2	2.8.E	Global cell for IOCS line.
XQ3	2.9.0	Global cell for IOCS line.
XQ4	2.9.2	Global cell for IOCS line.
XQ5	2.9.4	Global cell for IOCS line.
XQ6	2.9.6	Global cell for IOCS line.
XQ7	2.9.8	Global cell for IOCS line.
XQ8	2.9.A	Global cell for IOCS line.
XQ9	2.9.C	Global cell for IOCS line.
XQ10	2.9.E	Global cell for IOCS line.
XQ11	2.A.0	Global cell for IOCS line.
XQ12	2.A.2	Global cell for IOCS line.
XQ13	2.A.4	Global cell for IOCS line.
XQ14	2.A.6	Global cell for IOCS line.
XQ15	2.A.8	Global cell for IOCS line.
XQ16	2.A.A	Global cell for IOCS line.
XDIA5	2.A.C	Setup point "Above"
YRESRV	2.A.E	TONS auxiliary cell
XTMERK	2.B.0	TONS global cell
XCHGA	2.B.2	MP device type
XCHCC	2.B.4	Cylinder number
XCHSK	2.B.6	Sector number
XCHHD	2.B.8	Kop number
XCHVER	2.B.A	Trial indicator
XCH 1	2.B.C	Working cell 1
XCH 2	2.B.E	Working cell 2
XCH 3	2.C.0	Working cell 3
PUFF	2.C.2	12 bytes header for disk
XULAD	2.C.A	Working cell
XUCNT	2.C.C	Working cell

Symbol	Address	Meaning
XADDR	2.C.E	Block address
ANF 1	2.D.0	Start of bits 17 to 18.
ZMZEND	ANF1+1.1.E	Last line of the system nucleus.
YZGRA	ZMZEND+2	Contains the entry address for signing off SP.
YZASTP	ZMZEND+4	Contains the entry address for the job limit time test program.
YZASXI	ZMZEND+6	ZAS index
YZASRT	ZMZEND+8	Return address from devices to ZAS.
YZASPZ	ZMZEND+10	Present buffer pointer.
YZASPL	ZMZEND+12	Present buffer length.
YZASCO	ZMZEND+14	Error messages or addition value for the job limit time.
IOGLOB	ZMZEND+16	Start of the device module global cells.
IOGLOOO	IOGLOB+0	Output table 2 E/A 1801 (assigned by card reader 0090 and paper tape reader).
IOGLOO1	IOGLOB+2	Output table 1 E/A 1801 (assigned by SS).
IOGLOO2	IOGLOB+4	"Assigned" flag line (TC 400).
IOGLOO3	IOGLOB+6	Flag for console change, Not Close (DRY).
IOGLOO4	IOGLOB+8	Global cell for disk drive
IOGLOO5	IOGLOB+10	Global cell for disk drive
IOGLOO6	IOGLOB+12	Global cell for disk drive
IOGLOO7	IOGLOB+14	Global cell for disk drive
IOGLOO8	IOGLOB+16	Global cell for disk drive
IOGLOO9	IOGLOB+18	Global cell for disk drive
IOGLO10	IOGLOB+20	Global cell for disk drive
IOGLO11	IOGLOB+22	Global cell for disk drive
IOGLO12	IOGLOB+24	Global cell for disk drive
IOGLO13	IOGLOB+26	Global cell for disk drive
IOGLO14	IOGLOB+28	Global cell for magnetic tape transport
IOGLO15	IOGLOB+30	Global cell for magnetic tape transport
IOGLO16	IOGLOB+32	Global cell for magnetic tape transport
IOGLO17	IOGLOB+34	Global cell for magentic tape transport
IOGLO18	IOGLOB+36	Global cell for magnetic tape transport
IOGLO19	IOGLOB+38	Global cell for magnetic tape transport
IOGLO20	IOGLOB+40	Global cell for magnetic tape transport
IOGLO21	IOGLOB+42	Global cell for magnetic tape transport

15 File Assignment List

The file assignment list occupies 321 bytes, so that up to 32 files may be entered. Each file entry occupies 10 bytes (32 x 10 = 320 bytes).

- Arrangement of the file assignment list

	Byte	
	1	Contains the quantity of files that have so far been entered (i.e. opened).
1st file entry	2 3 4 5 6 7	These six bytes contain the name of the first file opened.
	8 9 10	These three bytes contain the address of the file control block allocated to the first file which has been opened.
	11	This byte states how often the first file has been opened. Required for TACOS
2nd file entry	12 13 14 15 16 17	These six bytes contain the name of the second file opened.
	18 19 20	These 3 bytes contain the address of the assigned file control block (FCB) for the second file opened.
	21	This byte states how often the second file has been opened. Required for TACOS.
	. . . . .	
	312 313 314 315 316 317	These six bytes contain the name of the 32nd file opened.
	318 319 320	These three bytes contain the address of the file control block for the 32nd file.
	321	This byte states how often the 32nd file has been opened End of file assignment list.

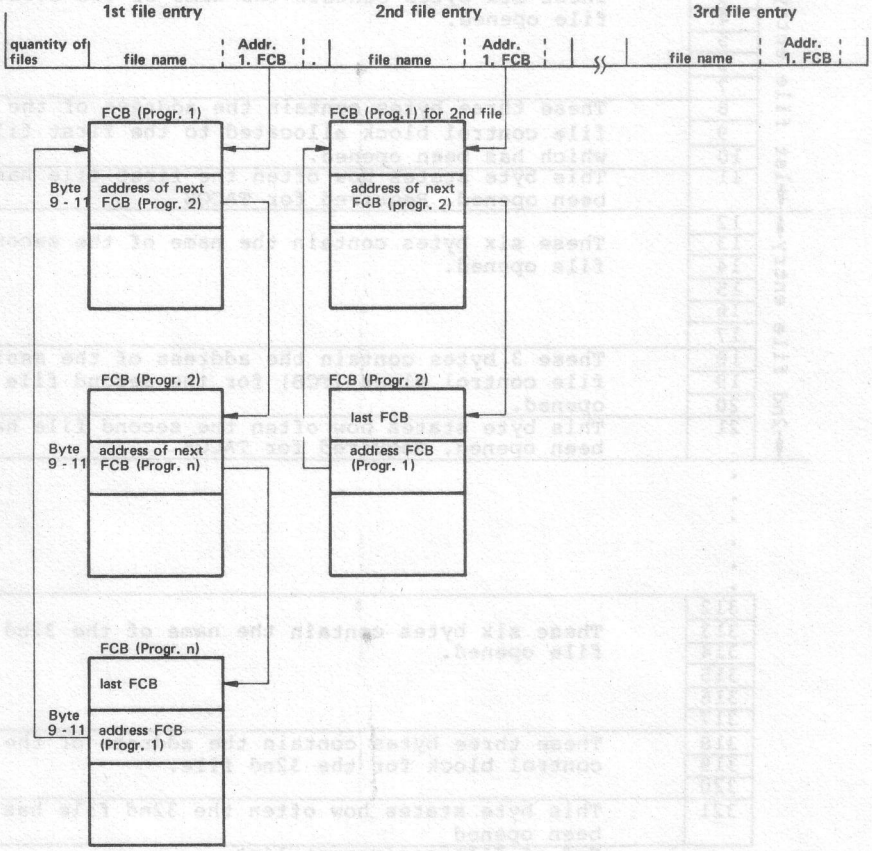
The file assignment list is managed by OPEN and CLOSE.

Each file is entered once only.

If more than one program addresses a file at the same time, the file control blocks for the file in question are chained to one another. The chaining address is in bytes 9 to 11 of the appropriate FCB. The chaining address of the last FCB points to the start address of the first (ring chain).

Example:

File Assignment list



FCB chain from the first file

16 FCB Assignment List

The FCB is assigned as follows:

- Present data

Current Byte No.	Quantity of bytes	Content
1	1	Test byte
2	1	Program recognition
3 - 5	3	Archive number
6 - 8	3	AA FILLAB 3 dual address characters
9 - 11	3	AA fo the FCB in the other program
12 - 14	3	AA of the device control field
15 - 17	3	Error address 3 dual address chars
18 - 20	3	AA user 1) AREA 3 dual address chars
21 - 23	3	AA 13 block in the user area 3 dual address characters
24 - 26	3	AA ACC-AREA 3 dual address characters
27 - 29	3	AA IKE field 3 dual address chars
30 - 32	3	AA WORKAREA 3 dual address characters
33	1	Flag byte I
34	1	Flag byte II
35 - 40	6	Present data record address CC/H/R/ZZ
41 - 46	6	Present 13 overflow address CC/H/R/ZZ
47 - 49	3	Present record no. 3 dual ADR chars
50	1	Access mode
51	1	Bit mask
52	1	Free

- File label

Current Byte No.	Quantity of bytes	Content
53 - 58	6	File identifier
59	1	File record disk address, H/R
60	1	File record disk address (keyed)
61	1	Orga-byte I
62	1	Orga-byte II
63 - 64	2	Quantity of records per track
65	1	Quantity of records per block
66	1	Quantity of sectors per block (data file)
67 - 68	2	Record length in bytes
69 - 70	2	Block length in bytes
71 - 73	3	AA Data file CC/H
74 - 76	3	EA Data file CC/H
77 - 82	6	Largest written data record CC/H/R/ZZ
83 - 86	4	Error bytes
87 - 89	3	Release date
90 - 92	3	Creation identification

- Index sequential files

Current byte no.	Quantity of bytes	Content
93 - 94	2	Quantity of indices per track 13
95	1	Quantity of indices per block 13
96 - 97	2	Quantity of indices per block 12
98	1	Quantity of sectors per 13-block (SF 1-6)
99	1	OB length
100 - 101	2	I 3 block length
102 - 104	3	AA I1 area 0-1/Byte no.
105 - 106	2	AA I2/I3 area CC
107 - 112	6	AA of the first delete label, CC/H/R/ZZ
113 - 115	3	EA I1 area 0-1/Byte no.
116 - 118	3	Largest I1 status 0-1/Byte no.
119 - 120	2	AA of the limit order term in byte 9.
121 - 122	2	Largest I1 status ZZ
123 - 128	6	Largest I3 status CC/H/R/ZZ
129 - 134	6	Largest overflow pool status CC/R/H/ZZ
135 - 137	3	EA of the I2/I3 area CC/H

- With CONNECT (chaining)

Current byte no.	Quantity of bytes	Content
138 - 143	6	File identifier 1
144 - 149	6	File identifier 2
150 - 155	6	File identifier 3
156 - 161	6	File identifier 4
162 - 167	6	File identifier 5
168 - 173	6	File identifier 6
174 - 179	6	File identifier 7
180 - 182	3	Chaining of all identical opened files
183 - 185	3	Chaining of chain files within one program

This data must be protected against erasure or overwriting. Furthermore, no alterations should be made by the user, as this could give rise to discrepancies and errors in the disk processing.

17 Display Parameter Field

The correspondence between the operating system and the "Display" channel program is not achieved by means of general I/O instructions, but by the handling of system and user parameters which is necessary similarly to a device control field, 18 bytes long, located in the byte area.

The parameter field is loaded by the operating program and the jobs that have been entered are scanned by the channel program at every bell call (every 8 ms).

17.1 Assignment of the Parameter Field

Byte	Meaning
0	Left-hand flag byte Bit 8 (M15) = 1, New S-job pending or being processed. Bit 8 (M15) = 0, S-job ended, or no job currently pending.
1 - 2	not used
3	I/O code **
4 - 6	not used
7 - 9	M-address *
10 - 11	N value (without taking account of the preliminary byte) N value ≙ (Byte 10) • 256 * (Byte 11) 256 = Value of byte 10
12 - 13	Address of the global cell for Not Close
14 - 17	not used

* M address

The M address indicated the start address of the source field.

Arrangement of the source field:

The source field consists of both the preliminary bytes and the text. The length of the text is specified by the N value.

1st preliminary byte	2nd preliminary byte	Text
lines 0 - 10	columns 0 - 79	The length is specified by N.

** I/O Codes

The following I/O codes are used in the parameter field.

Code	Meaning
0.0	N character display on the screen and in the S buffers. These are only displayed if the console position is in the system status. The address above which the characters are to be shown is to be found in the preliminary bytes.
0.1	Roll over: N characters are moved upwards or downwards. A maximum of 80 N characters are transferred to the line that has thus become free (this line = the M address of the first leader byte). The first leader byte contains the start address of the movement, the second its end address. <VB1> > VB1>≙ Roll up <VB1> < VB2>≙ Roll down <VB1> ≙ VB2>≙ No movement (dummy instruction) Example: VB1 = 5, VB2 - 2 Operation: <Line 3> →Line 2 <Line 4> →Line 3 <Line 5> →Line 4 N-Character →Line 5 Roll over only takes place when the console position is in the system status.
0.2	Clear the screen. No interpretation of M address, N value or leader bytes. From line 0, column 0 880 or 1840 blanks are always transferred into the S-buffer and on to the screen. This only works when the console position is in the system status.
0.3	Change console. The operating system informs the console by means of this instruction that another display is to become console. The device number of the new display console is codified as an N value in the parameter field.
0.4	Not close. The program is aborted, either because of a PRG 21 applied by the user program or on the instructions of the operating program. The device number is codified in the N value. This instruction interrupts current GET instruction and the screen and A-buffer are cleared.
0.5	Not close. Same function as I/O instruction 0.4, except that the "clear" function is not performed. The device number is codified in the N value.

18 "Operating System" Common Area

The Common Area, covering 326 bytes, operates as a communications area for the system between foreground and background.

18.1 Arrangement of the Common Area

Symbol addr.	Absol. addr.	Meaning
CMATYP	0	Description of machine type /2 = 3.2, /4 = 3.4, /6 = 3.6, /5 = 3.5 (reset /4).
CPARDI	2	Start ADR of parameter field display.
CFGXDA	5	Memory area for the content of XDA (X5), foreground.
COBELG	8	Length of the user common area, if this information is available.
COBEAA	11	Start address of the user common area, if available.
COBEME	14	2 byte flag field for user common area. For flag assignment see section 18.2
COMER 1	16	2 byte flag field for various system messages. For flag assignment see section 18.3.
COMRE 1	18 . . . 21	Not used at present.
COMERLO	22	2 byte flag field for the logic file. For flag assignment see section 18.4.
CFCBL 0	24 . . . 82	FCB for the logic file.

Symbol addr.	Absol. addr.	Meaning
CIKEFE	83 . . 86	IKE Field for the log file.
CIOARE	87 . . 169	IO-AREA (buffer) for the log file.
CARCH	170	Archive number of the disk on which the log file is to be found.
COVERL	173	IO-CODE (Start of the IO field overlay loader).
COVMER	175	1 flag byte for the overlay loader. For flag assignment see section 18.5.
COVFEL	176	Error code (the error code refers to the loading process).
COVAAG	180	Start address of the device control field of the disk station on which the overlay phases are located.
COVZIL	183	Target address in the memory, above which the overlay phase is to be entered.
COVLNG	186	Block length of the overlay phase to be loaded.
COVPLA	188	Disk address of the overlay phase to be loaded.
COVRST	192	Code byte/Sector factor.

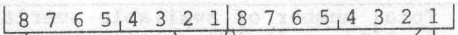
NIXDORF COMPUTER Kundendienst

Symbol addr.	Absol. addr.	Meaning
COVPGN	194	Device number of the system disk.
COVPGA	195	Device type of system disk DA 0/1 = 2, DA 3 = 4.
COVBJT	196	Control byte for overlay loader, see section 18.6.
CANZPA	197	Total number of available overlay partitions in the overlay storage area (1 partition = 512 bytes).
CLABYT	198	The partition from which the search for free paritions begins is always in this byte.
PARMAT	1.9 . . . 230	Partition group - 1 1 byte is assigned per partition. Entries are only made in the byte of the first partition required by the "N"s in a module. 8 7 6 5 4 3 2 1 Bit Bits 1–6 → Module no. Bit 7 → "1", the overlay module is operating in BG. Bit 8 → "1", the overlay module is operating in FG.
MODMAP	231 . . . 291	Module parameter group 2 bytes are assigned per module. Each time a system is opened, the number of partitions necessary for each module is calculated and entered in the MODMAP. If a module is called, the number of the first partition assigned by the module is entered. [Quantity of required partitions] [No. of the first partition assigned] No. of the first partition assigned → This byte is the criterion for determining whether the module is transient or resident. If byte ≠ 0, the module is still resident.
CPARET	292 . . 324	Save field for PARMAP, if overlay is too small.
COMLNG	325	Length of the common area (326 bytes)

18.2 COBEME Flag Assignment (Common Area 2 bytes flag field)

COBEME 0
Adr. SYX
XCOM, 14

COBEME 1
Adr. SYX
XCOM, 15



= "1",
Use of the common area is barred to TACOS and FG. (It is also possible to inhibit TACOS in signle).

= "0",
Use of the common area is not barred to TACOS or to BG.

= "0" and Bit 1 = "0",
Any program unit entitled to use the common area may do so without any restriction whatsoever.

≠ "0" and bit 1 = "1",
Use of the common area is barred to TACOS and only permitted for TRP; the TRP number is in bits 1 to 8.

= "0 1",
The common area is being used by the FG.

= "1 0",
The common area is being used by the BG.

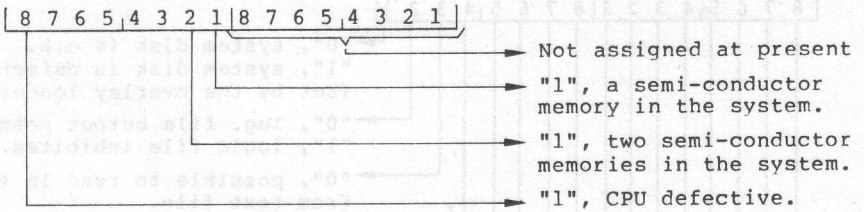
= 1 1",
The common area is being used by Single or TACOS.

= "1",
The common area is being processed (is active).

= "0",
The common area is not being processed.

18.3 COMER 1 and COMER 1.1 Assignment (2 byte flag field for various system functions)

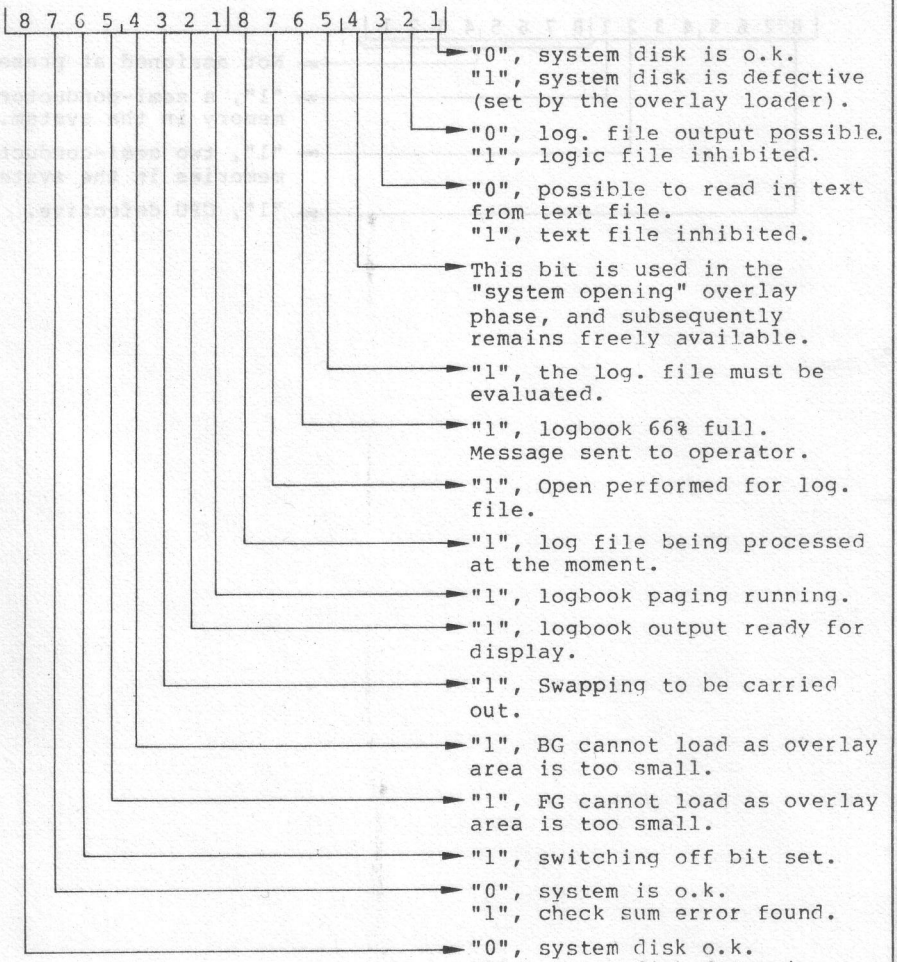
```
COMER 1              COMER 1,1
Adr. SYX             Adr. SYX
XCOM, 16             XCOM, 17
```



18.4 CMERL 0 and CMERL 0.1 Flag Assignment (2 byte flag field for the logic file)

CMERL 0
Adr. SYX,
XCOM, 22

CMERL 0,1
Adr. SYX,
XCOM, 23



18.5 COVMER Flag Assignment (1 flag byte for the overlay loader)

```
COVMER
Adr. SYX,
XCOM, 175
```

8	7	6	5	4	3	2	1

18.6 COVBIT Flag Assignment (Control byte for overlay loader)

```
COVBIT
Adr. SYX,
XCOM, 196
```

8	7	6	5	4	3	2	1

- Bit 1: "1", Overlay loader is active.
- Bit 2: "1", loading or unloading of the 14 overlay area partitions (Swapping).
- Bit 3: "1", FG parked.
- Bit 4: "1", BG parked.

For notes

19 Overlay Area

Certain transient operating programs are kept in a particular storage area of the main memory - the overlay area - for as long as they are required.

This has a size of 8 k bytes. Its start address in in X425.

Normally, that is to say when TIOCS is not being used, the overlay area occupies H.A. blocks 3.B and 3.C. When TIOCS is being used it moves downwards.

19.1 Partitioning of the Overlay Area

The overlay area is split into partitions of 512 bytes. This means that 16 partitions each fo 512 bytes are available to receive program modules.

19.1.1 Available Program Modules

Module phase no.	Module no.	Function
9500	0	System opening. } These modules are not based in the overlay area. They are loaded in the user area, i.e. they are positioned in the memory relative to XLAD.
9510	1	Ready stop.
9520	2	Job control.
9530	3	Free
9540	4	Receive the following operating instructions, 1: CANCEL, TEST, MS, ASSGN, RELSE, LOAD, TEST KOMMAND)
9550	5	Receive the following operating instructions, 2: DEFINE, SET, LOADCL, UNLOAD)
9560	6	Receive the following operating instruction, 3: DISPLAY.
9570	7	Device control printing.
9580	8	Loading program
9590	9	PRG
9600	10	TST 1
9610	11	TST 2
9620	12	IOC (IOCS OPEN/CLOSE).
9630	13	OWF
9640	14	IFB (IOCS error handling)
9650	15	Core dump
9660	16	BK4 (user communication 4)
9670	17	BK5 (user communication 5), turn over sheets in log book.
9680	18	CD 8 (CLOSE 8870)
9690	19	OD 8 (OPEN 8870)
9700	20	TACOS OP/C (OPEN/CLOSE)
9710	21	RAP LOADER
.		
.		
.		
9790		
9800		
.		
.		Future RAP phases
.		
9990		

19.2 Loading the Overlay Modules

The modules to be loaded are entered consecutively in the overlay area. When it is proposed to load a module, its length must first be ascertained and divided by 512, to find out how many partitions it occupies in the overlay area. Next, a check is made to see whether enough partitions are still available in the overlay area. The whole area is searched once to find this out.

If a free area is found anywhere, the new phase is entered there and processed. If insufficient space is available, there is either a change of level (only possible with double operation), or 7 k bytes are transferred from the overlay area to the disk so as to obtain enough storage space for the new phase to be accepted (single operation). Once the new overlay phase has been processed the phases transferred to the disk are loaded back into the overlay area and processed further.

In the case of double operation, there is first a change of level, in the hope that the other level will be in a position to process the overlay phase. If, however, the other level calls in a further overlay module, which has to be loaded, 7 k bytes are once again transferred to the disk to provide room for the new overlay module. A flashing asterisk in the status line indicates that the appropriate level is waiting for a free space in the overlay area.

19.3 Ascertaining the Overlay Module Number in the event of a Failure

Example: The system continues to process only those instructions in overlay area 3.C.2.5.6 to 3.C.2.15.8.

Task: To find out in which overlay module the system is working.

Method: To locate the module, its number must first be ascertained. This is always in the first partition of the module, at the start address of the partition + 1.15. As the size of a partition is always 512 bytes, the address gap between two partitions is always 2.0.0. Thus, the address of the module number will always be found at one of the following addresses.

19.3.1 Memory Distribution of an Overlay Partition

Address	Byte address	Quantity	Meaning
3.B.0.0.0	0 1 . . 11	1 2	These bytes are used by IOCS. They contain the disk address of the loaded overlay module.
3.B.0.0.B	12	1	Index ADR = relative start ADR of the loaded module.
	13 14 15		Blank bytes
	1,0 1,1 1,2 1,3 1,4 1,5 1,6 1,7	1 2 3 4 5 6 7 8	Date = creation date of the overlay module.
	1,8 1,9	1 2	Version no. of the loaded overlay module.
	1,10 1,11	1 2	Length of the loaded overlay module.
	1,12 1,13 1,14	1 2 3	Module name = module abbreviation
3.B.0.1.F	1,15	1	Module number
	2,0 2,1	1 2	Disk address for DA 0 and 1. Byte 1 = head; byte 2 = sector cylinder always zero.
	2,2 2,3	1 2	Free
	2,4 2,5		Disk address for DA 3. 1st byte, bits 1 to 5 = head, bits 6, 7 and 8 = cylinder. 2nd byte = sector no.
3.B. 0.1.15 3.B. 2.1.15 3.B. 4.1.15 3.C. 0.1.15 3.C. 2.1.15 3.C.14.1.15			Address where a module number may be found.

With reference to the example on page 121:

As only macro-instructions are processed at addresses 3.C.2.5.6 to 3.C.2.15.8, addresses 3.C.2.0.13 to 3.C.2.0.15 (3 blank bytes) are the first places to look to see whether any zeros have been entered.

If none have, the search must continue in the next partition down (3.C.0.0.12 to 3.C.0.0.15). If there are no blank bytes here either, the search must be continued back as far as is necessary to find the three being sought.

Let us say, for the sake of this example, that the 3 blank bytes are at 3.C.0.0.12 to 3.C.0.0.15. To be able to find the module number, 0.1.15 must be added to the start address of the partition in question (3.C.0.0.0). There must be a module number between 0 and 30 at this address (3.C.0.1.15). If there is no value between 0 and 30, the start of the module being sought is not in this partition. The search must then be continued by repeating the procedure described above.

Once a module number has been found, it can be used to find out which module has been allocated that number. If the absolute address is required, 3.C.0.1.12 (= index address of the overlay module) is to be subtracted from addresses 3.C.2.5.6 to 3.C.2.15.8.

Thus:
$$\begin{array}{rl} & 3.C.2.5.\ 6 \\ - & \underline{3.C.0.1.12} \\ & 0.0.2.3.10 \end{array} = \text{absolute address in the relevent overlay module.}$$

For notes

NIXDORF COMPUTER
Kundendienst

Change Report

To:

NIXDORF COMPUTER AG
TKD - Ausbildung und Information

Fürstenallee
4790 Paderborn

From:

Title of Manual: ....

Order Number: ....

Changes requested/errors noted: ....

....

....

....

....

....

....

Sketches:

Date: ....

Signature: ....

Please observe back of page.

Ä.M. eingegangen am:	Bearbeiter:	Ä.M. erledigt am:

NIXDORF COMPUTER
Kundendienst

Please note:

This change report enables you to report any errors, required changes, or corrections.

You will help both yourself and us when making use of this form.

Detach the change report from your manual and forward it to the given address if required.

If possible, attach the incorrect page(s), with errors marked, to the change report.

Please contact the above address for further forms.

When a manual is corrected or changed, the edition number and the date of issue printed on the cover sheet is also changed. In addition, any changed page is given the same date of issue as printed on the cover sheet.