

GCOS 66 SOFTWARE DESIGN SPECIFICATION

BUFFER MANAGEMENT

Written by: C. Coflin

Approved by: _____
J. S. McDade
Advanced File System Design

Issue Date: December 15, 1977
Revision Date: June 1, 1978

CONTENTS

	Page
Section I BUFFER MANAGEMENT	
1.0 General Description	1-1
2.0 Interface to Buffer Management	1-3
2.1 Buffer Management Control Structures	1-4
2.1.1 Buffer Management Command Block (BMCB)	1-4
2.1.2 Modification Control List	1-9
2.1.3 Modification Descriptor List	1-12
2.2 Buffer Management Functions	1-13
2.2.1 Get Control Interval (GETCI)	1-13
2.2.2 Modify Control Interval (MDFCI)	1-15
2.2.3 Change Control Interval Attributes (CCIAT)	1-17
2.2.4 Flush Buffers (FLUSH)	1-19
2.2.5 Force Write of Control Interval (FORCE)	1-20

ILLUSTRATIONS

	Page
Figure 1.2-1 Buffer Management Control Block	1-4
Figure 1.2-2 Modification Control List Entry	1-9
Figure 1.2-3 Modification Descriptor List	1-12

SECTION I
BUFFER MANAGEMENT

1.0 General Description

Buffer Management is a centralized shared component of GCOS66 and is the focal point for all buffer handling and access-time file security and integrity checking.

Different algorithms for managing buffers are required to efficiently support different applications. These algorithms determine such buffering characteristics as how long a control interval is allowed to remain in a buffer before that buffer is used for a different control interval. Thus Buffer Management will apply one algorithm to a group of buffers. This group of buffers is called a buffer pool.

When a file is opened, it is mapped onto one, and only one, buffer pool. One buffer pool, however, may have one or more files mapped to it. A file that is sharable must be mapped to a shared buffer pool. That is, a pool whose buffers exist in a shared working space. A non-sharable file may be mapped to either a shared or non-shared buffer pool.

| The function which opens the file returns two parameters to
| the caller: an entry descriptor with which to call Buffer
| Management and a file identifier which must be input to
| Buffer Management to identify the file to be accessed.

Buffer Management provides five functions for accessing the file.

1. The Get Control Interval (GETCI) function provides the caller with addressability to the requested CI. If the requested CI is not already in a buffer, an available buffer is located (possibly triggering one or more WRITE operations) and the CI is read into the buffer. A segment descriptor framing the CI is returned to the caller. If the file is sharable, the file and/or control interval will be enqueued to control concurrent access conflicts.

2. The Modify Control Interval (MDFCI) function modifies one or more fields of a CI as specified by the caller. If the file is protected, both before and after images of the modified fields are journalized. If the control interval has been enqueued for shared usage, Buffer Management enqueues the CI for exclusive use.
3. The Change Control Interval Attributes (CCIAT) functions allows the caller to change the state of permissions to a CI:
 - the reservation of the CI may be changed from read (shared usage) to update (exclusive usage).
 - addressability to the CI may be extended by locking the CI; the buffer containing a locked CI may not be reused.
 - a CI that was previously locked may be unlocked
4. The Flush Buffers (FLUSH) function writes all buffers for the specified file that contain control intervals that were modified by the caller.
5. The Force Control Interval (FORCE) function forces the immediate writing of a modified CI rather than allowing the buffering algorithm to determine when the buffer should be written.

2.0 Interface_to_Buffer_Management

There is one entry point into Buffer Management for all functions. This entry is executed by a CLIMB instruction that references an entry descriptor in the caller's linkage segment. This entry descriptor is created by the OPEN file function.

After the CLIMB to Buffer Management is executed, the parameter stack must contain the following segment descriptors:

- descriptor 0 (all functions)
frames the Buffer Management Command Block (BMCB) which contains the input and output parameters.
- |
|
| - descriptor 1 (GETCI function only)
| frames an area of descriptor storage in which Buffer Man-
| agement will store the segment descriptor that frames the
| requested CI.
- descriptor 1 (MDFCI function only)
frames the Modification Control List which describes each modification to be made to the Control Interval.
- descriptor 2 (MDFCI function only)
frames the Modification Descriptor List which is an array of segment descriptors each of which describe a segment that contains one or more of the source fields which will be moved to the CI during the modification.

The Buffer Management function executed depends on the setting of index register 0.

XR0 = 1	GETCI
= 2	MDFCI
= 3	CCIAT
= 4	FLUSH
= 5	FORCE

* 2.1 Buffer_Management_Control_Structures

2.1.1 Buffer_Management_Command_Block_(BMCB)

The BMCB is described by the first descriptor on the parameter stack when Buffer Management is called. It is a block created by the caller of Buffer Management which contains the parameters that are input to Buffer Management, as well as the output parameters Buffer Management will return to the caller.

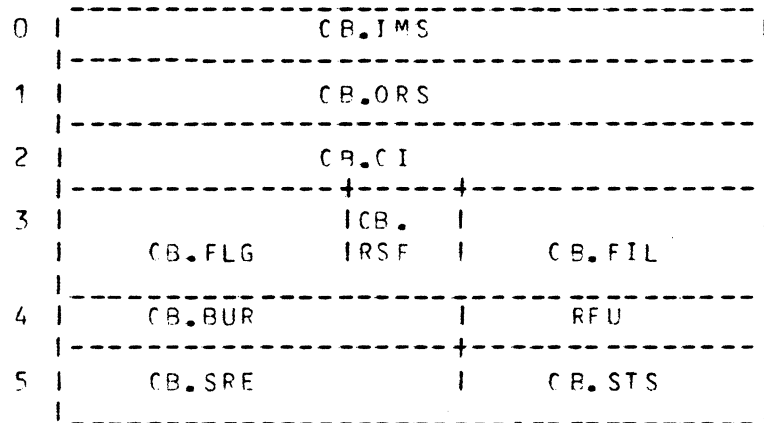


Figure 1.2-1. Buffer Management Control Block

Word_0

Bit 0 (36) : CB. IMS
 Immediate Status
 The format of this word conforms to the return code standard.

bits 0-17 : segment number
 18-23 : entry point number
 =1 GETCI
 =2 MDCCI
 =3 CCIAT
 =4 FLUSH
 =5 FORCE
 24-35 : return code
 =0 normal
 =1 Concurrent Access Control Conflict
 =2 input error
 =3 I/O error
 =4 logic error

Word_1

Bit 0 (36) : CB.ORS
Original Status
The format of this word conforms to
the return code standard.

Word_2

Bit 0 (36) : CB.CI
Control Interval number
This number is used by Buffer Manage-
ment to map to a mass storage address.
The first CI number in a file is CI
zero. All control intervals for a given
file are the same size which is an
integral number of sectors for files on
disk

Word_3

Bit 0 (16) : CB.FLG
Flags

0 : Update_Flag
This bit is on if update permission to
the control interval is desired.

1 : Conflict_Flag
This bit is on if the caller wishes
control returned in case of a concur-
rent Access Control conflict, rather
than waiting until the CI is available.

2 : Lock_Flag
This bit is on if the caller wishes to
lock (extend addressability) to the
current buffer. The buffer will not be
reused until it is unlocked by the
caller.

3 : Unlock_Flag
This bit is on if the caller wishes to
unlock a previously locked buffer.

4 : New_Flag
This bit is on when a CI is to be
created. An empty buffer is returned
rather than reading the CI from the
file.

- 5 : Sequential_Flag
This bit is on when the caller wishes to advise Buffer Management that the control intervals for this file are being read/written sequentially. This allows Buffer Management to optimize the buffering techniques for sequential access.
- 6 : Asynchronous_Flag
This bit is on when the caller desires to execute asynchronously with the I/O. To insure that the function has been completed, the caller must re-execute the function.
- 7 : No_Currency_Flag
This bit is set when the current and locked buffers for the caller are to be released.
- 8 : Physical_Journalization_Flag
This bit is set when the after images for this file and user are to be physically written to the after journal as part of this function.
- 9-15 : RFU
- 16 (2) : CB.RSF
Residency Factor
The residency factor is used to control the re-use of buffers. Buffers containing low residency CI's will be reused before those containing medium or high residency CI's.
- = 00 low residency
= 01 medium residency
= 10 high residency
- 18 (18) : CB.FIL
File identifier
This is the 18 bit file identifier that is returned to the caller by the OPEN file function.

Word_4

Bit 0 (18) : CB.BUR
BUR identifier;
This identifier is returned by the
GETCI function and is used by other
functions to identify the buffer.

18 (18) : RESERVED

Word_5

Bit 0 (18) : CB.SRE
Sub-resource reservation entry;
If the file reservation type is shared
at the sub-resource (i.e., control
interval) level, then the requested
control interval is enqueued and the
SRE (sub-resource reservation entry)
identifier is returned. If the file
reservation type is either exclusive or
shared at the file level, then a zero
SRE is returned.

18 (18) : CB.STS
Detailed Status (decimal values)
= 0 - 4 Normal Termination
0 function complete
1 asynchronous operation not
complete
2 last CI of file
3-4 reserved
= 5 - 9 Concurrent Access Control Con-
flict
5 time-out waiting for CI
6 deadlock
7-9 reserved

= 10-29	Input Error
10	illegal command in x0
11	illegal CB.CI
12	illegal CB.FIL
13	illegal CB.BUR
14	illegal CB.SRE
15	ILLEGAL MC.DO (DESTINATION OFFSET IN MODIFY LIST)
16	ILLEGAL MC.SO (SOURCE OFFSET IN MODIFY LIST)
17	ILLEGAL MC.FIL (FILL CHARACTERS)
18	ILLEGAL MC.IDX (DESCRIPTOR INDEX)
19-29	RESERVED
= 30-49	I/O ERROR
30	READ ERROR
31	WRITE ERROR
32	illegal seek address
33-49	reserved
= 50-69	Logic Error
50	too many buffers locked
51	no modification permission
52	buffer not locked
53	buffer not current
54	buffer not modified
55	Buffer Mgt internal error
56	No buffer is available
57	No BUR is available
58	Buffer not current or locked
59-69	reserved

2.1.2 Modification_Control_List

The Modification Control List is described by the second descriptor on the parameter stack when Buffer Management is called. It is a list created by the caller of Buffer Management which describes the modifications to be made to a buffer. This list is composed of a set of contiguous entries, where each entry describes one modification to the buffer. The format of each entry is shown in Figure 1.2-2.

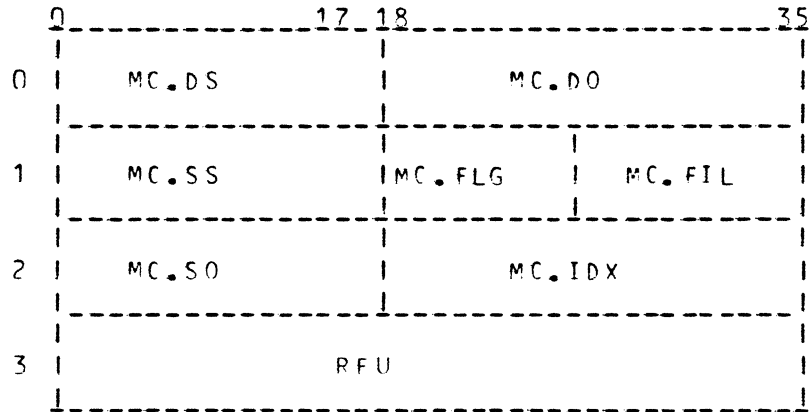


Figure 1.2-2. Modification Control List Entry

Word_0

- Bit 0 (18) : MC.DS
Destination Size
The length in 9-bit bytes of the field in the buffer to be modified.
- 18 (18) : MC.DO
Destination Offset
The offset from the beginning of the buffer in 9-bit bytes of the field to be modified. The offset of the first byte in the buffer is 0.

Word_1

Bit 0 (18) : MC.SS
Source Size
The length in 9-bit bytes of the source field for the modification.

18 (9) MC.FLG
Flags

18 Last_Entry_Flag
This bit is on if this is the last entry in the Modification Control List.

19 No_Before_Flag
This bit is on if a before image of the field to be modified is not to be taken.

20 No_After_Flag
This bit is on if an after image of the modified field is not to be taken.

21 No_Move_Flag
This bit is on if the modification of the buffer is not to be performed. This flag will be used to journalize the field and not perform the modification.

22 Buffer_to_Buffer_Flag
This bit is on when the source field is in the destination buffer. The source offset (MC.S0) is an offset from the beginning of the buffer.

23 Right_to_Left
This bit is on when a right to left move is desired. When this bit is off, a left to right move is performed.

24-26 Reserved.

27 (9) : MC.FIL
Fill Character
This field contains the octal value to be used for a fill character in the EIS move. Legal values for this field are:

000 - binary zero
060 - ASCII zero
040 - ASCII blank

Word_2

- Bit 0 (18) : MC.S0
Source Offset
The offset from the beginning of the source segment in 9-bit bytes of the source field for the modification. The offset of the first byte in the segment is 0.
- 18 (18) : MC.IDX
Descriptor Index
The descriptor number of the segment descriptor in the Modification Descriptor List that described the source segment. The number of the first descriptor is 0, the second descriptor is 1, etc.

2.1.3 Modification_Descriptor_List

The Modification Descriptor List is described by the third descriptor on the Parameter Stack when Buffer Management is called. It is a list of segment descriptors which describe the segment(s) containing the source field(s). Each entry in the modification Control List references one of the descriptors to identify the source segment.

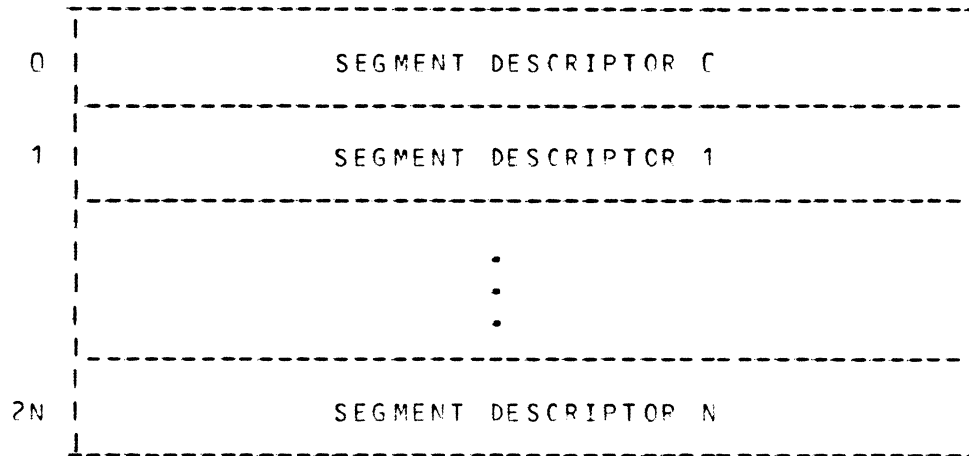


Figure 1.2-3. Modification Descriptor List

2.2 Buffer_Management_Functions

2.2.1 Get_Control_Interval_(GETCI)

This function retrieves the requested Control Interval and returns its address.

Input_Parameters

XRD = 1 GETCI function

BMCB

CB.CI The number of the requested Control Interval.

CB.FIL File identifier; output by the CPEN file function.

CB.RSF Residency factor to be applied to this CI.

CB.FLG Flags

- Update On if the CI is to be modified.
- Conflict On if control is to be returned in case of a concurrent Access Control conflict.
- Lock On if the CI is to be locked.
- New On if an empty buffer is desired in which to create the CI.
- Sequential On if the Control Intervals of the file are being accessed (or created) in sequential order.
- Asynchronous On if access to the CI is not desired at this time.

Output_Parameters

1 Segment_Descriptor_of_Requested_CI

BMCB

CB.IMS Immediate Status
(see definition of BMCB for values)

CB.ORS Original Status

CB.BUR Identifies of the Buffer/User Relationship (BUR) associated with this request. This identifier is an input parameter to some of the other functions.

CB.SRE Identifier of the sub-resources Reservation Entry used by Concurrent Access Control for the requested Control Interval. Certain Concurrent Access Control primitives require this identifier as an input parameter.

CB.STS Detailed Status
(see description of BMOB for values)

Rules

1. The first Control Interval in the file is CI zero. The file description contains the number of the last CI in the file that contains data. If a CI is requested which is greater than this last CI, the NEW flag must be set to indicate that the CI is to be created.
2. For synchronous I/O, upon a successful return, the requested CI becomes "current". A buffer containing a "current" or "locked" CI cannot be reused by Buffer Management. The CI will remain current until the next GETCI operation*. If addressability to a CI is desired following other GETCI functions, the CI must be locked via setting the lock flag when that CI is requested or by using the change CI Attributes (CCIAT) function once the CI is current.

*Note: Under certain options, the FORCE and FLUSH primitives will release the current CI.
3. A CI is considered modified and will be written to the file if either:
 - The update flag is set for the GETCI primitive,
 - the update flag is set for the CCIAT primitive, or
 - the MDFCI primitive is executed
4. If there is more than one CI contained in each buffer, modifying any of the CI's implies that all CI's in that buffer will be written to the file.
5. If the asynchronous flag is set, Buffer Management will read the CI if required, but will not wait for the I/O completion. On return, there is no "current" CI. When addressability to the CI is actually desired, another GETCI primitive for that CI must be executed with the asynchronous flag turned off.

2.2.2 Modify_Control_Interval_(MDFCI)

This function acquires update permission to the CI from Concurrent Access Control, if required, calls Integrity Control to journalize the "before images", performs the modification, and calls Integrity Control to journalize the "after images".

Input_Parameters

BMCB

CB.CI	The number of the CI to be modified
CB.FIL	File identifier; output by OPEN file function
CB.BUR	Identifier of the Buffer/User Relationship (BUR) returned by the GETCI function. Optional.
CB.SRE	Identifier of the sub-resource Reservation Entry returned by the GETCI function. Optional.

Modification_Control_List List describing each modification to be made to the CI.

Modification_Descriptor_List List of segment descriptors which frame the segments containing the source fields for the modifications.

Output_Parameters

BMCB

CB.IMS	Immediate status (see definition of BMCB for values)
CB.ORS	Original status
CB.STS	Detailed Status (see definition of BMCB for values)

*

Rules

1. A successful execution of the GETCI function for the CI to be modified must precede this function. In addition, the referenced CI must be "current" (i.e. the object of the last GETCI function) or "locked". Otherwise an error is returned.
2. The BUR identifier is an optional parameter to increase performance. If this parameter is zero or is invalid, a search will be performed to locate the buffer containing the object CI.
3. Each modification to the CI, described by one entry in the Modification Control List, is performed successively beginning with the first entry in the list. If an error is detected while processing entry N, the preceding N-1 modifications remain completed, while the remaining entries in the list (starting with N+1) are not processed.
4. The SRE identifier is an optional parameter to increase performance when changing the Control Interval reservation to "exclusive". If it is zero or invalid, a search will be performed to locate the reservation description for the object CI.
5. If the Update Flag was on when the GETCI function was executed for the referenced CI, the CI is already reserved for exclusive use and the call to Concurrent Access Control will not be performed.
6. The fill character used in the modification must be one of the following:
 - 000 - binary zero
 - 060 - ASCII zero
 - 040 - ASCII blank

2.2.3 Change_Control_Interval_Attributes_(CCIAT)

This function allows the caller to:

- request update permission to a CI,
- lock a CI, and
- unlock a CI

Input_Parameters

XRO = 3 CCIAT function

BMCB

CB.CI	The number of the CI whose attributes are to be changed
CB.FIL	File identifier; output by OPEN file function
CB.FLG	Flags;
- Update	On if update permission to the CI is desired.
- Conflict	On if control is to be returned in case of a Concurrent Access Control Conflict
- Lock	On if the CI is to be locked
- Unlock	On if the CI is to be unlocked
CB.BUR	Buffer/User Relationship (BUR) identifier output from the GETCI function; optional.
CB.SRE	Sub-resource Reservation Entry identifier output from the GETCI function; optional.

Output_Parameters

CB.IMS	Immediate Status (see definition of BMCB for values)
CB.ORS	Original Status
CB.STS	Detailed Status (see definition of BMCB for values)

Rules

1. A successful execution of the GETCI function for the referenced CI must precede this function. In addition the CI must be "current" (i.e., the object of the last GETCI function) or "locked". Otherwise an error is returned.

2. A CI may be locked more than once. It will not be released until it is unlocked the same number of times it was locked.
3. The BUR identifier is an optional parameter to increase performance. If this parameter is zero or invalid, a search will be performed to locate the buffer containing the object CI based on CB.CI.
4. The SRE identifier is an optional parameter to increase performance when the update flag is set. If it is zero or invalid, a search will be performed to locate the reservation for the object CI.
5. Update permission to the CI can be requested (update flag on) in conjunction with either locking the CI (lock flag on) or unlocking the CI (unlock flag on).

2.2.4 Flush_Buffers_(FLUSH)

This function forces all buffers of the file which have been modified by this user to be written to the file. Optionally, the current and all locked CI's may be released.

Input_Parameters

XRO = 4 FLUSH function

BMCB

CB.FIL File identifier; output by OPEN file function.

CB.FLG Flags;

- No Currency

On if the current and locked buffers are to be released.

- Physical Journalization

On if the after images are to be written to the journal file.

Output_Parameters

BMCB

CB.IMS Immediate status (see definition of BMCB for values)

CB.ORS Original status

CB.STS Detailed status (see definition of BMCB for values)

Rules

1. If the "no currency" flag is set, all CI's that are locked or current will be released. After the completion of this function, no other function may be executed until a successful GETCI.

2.2.5 Force Write of Control Interval (FORCE)

This function writes the buffer containing the referenced CI. Optionally the CI may be released if it is current or locked.

Input Parameters

XRO = 5 FORCE function

BMCB

CB.CI The number of the CI to be written.

CB.FIL File identifier; output from OPEN file function.

CB.FLG Flags;

- Asynchronous On if the caller wishes control returned before the write operation is complete.
- Sequential On if all modified CI's are to be written in the order of update up to and including the input CI number.
- No Currency On if the CI is to be released if it is current or locked.
- Physical Journalization On if the after images for this buffer are to be written to the journal file.

CB.BUR Identifier of the Buffer/User Relationship (BUR) returned by the GETCI function. Optional.

Output Parameters

BMCB

CB.IMS Immediate status (see definition of BMCB for values)

CB.ORS Original status

CB.STS Detailed status