INTERNATIONAL BUSINESS MACHINES CORPORATION
Data Processing Division
112 East Post Road
White Plains, New York, 10601
(U.S.A. only)

IBM WORLD TRADE CORPORATION
821 United Nations Plaza
New York, New York, 10017
(International)

GX28-6401-1
S 360-30

IBM System/360
Time Sharing System

TSS/360

Quick Guide for System Programmers

TSSS

ASSEMBLER

CONTROL BLOCKS

IBM

Systems Development Division

IBM System/360
Time Sharing System

TSS/360

General-Purpose Operating System

Quick Guide for System Programmers

TSSS

ASSEMBLER

CONTROL BLOCKS

Third Edition (June 1970)

This is a major revision of, and makes obsolete, X28-6401-0. This edition applies to Version 7, Modification 0, of IBM System/360 Time Sharing System, and to all subsequent releases until otherwise indicated in new editions or Technical Newsletters. Changes are periodically made to the specifications herein; before using this publication in connection with the operation of IBM systems, refer to the latest edition of *IBM System/360 Time Sharing System: Addendum,* Order No. GC28-2043 for the editions of publications that are applicable and current.

Text for this manual has been prepared with the IBM SELECTRIC® Composer.

Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form is provided with this publication for reader's comments. If the form has been removed, comments may be addressed to IBM Corporation, Time Sharing System/360 Programming Publications, Department 643, Neighborhood Road, Kingston, New York 12401.

TSS/360 System Reference Library
Introducing TSS/360, GC28-2048
Concepts and Facilities, GC28-2003
Data Management Facilities, GC28-2056
Assembler Language, GC28-2000
Assembler User Macro Instructions, GC28-2004
Assembler Programmer's Guide, GC28-2032
IBM FORTRAN IV, GC28-2007
FORTRAN IV Library Subprograms, GC28-2026
FORTRAN Programmer's Guide, GC28-2025
PL/I Reference Manual, GC28-2045
PL/I Library Computational Subroutines, GC28-2046
PL/I Programmer's Guide, GC28-2049
Linkage Editor, GC28-2005
Command System User's Guide, GC28-2001
Manager's & Administrator's Guide, GC28-2024
Operator's Guide, GC28-2033
Independent Utilities, GC28-2038
System Programmer's Guide, GC28-2008
System Generation and Maintenance, GC28-2010
Remote Job Entry, GC28-2057
Multiterminal Task Programming and Operation, GC28-2034
Terminal User's Guide, GC28-2017
System Messages, GC28-2037
Time Sharing Support System, GC28-2006
Master Index, GC28-2023
Quick Guide for Users, GX28-6400
Quick Guide for System Programmers, GX28-6401
Addendum, GC28-2043

TSS/360 Program Logic Manuals
System Logic Summary, GY28-2009
Resident Supervisor, GY28-2012
Task Monitor, GY28-2041
System Service Routines, GY28-2018
Dynamic Loader, GY28-2031
Access Methods, GY28-2016
Command System, GY28-2013
Program Control System, GY28-2014
Assembler, GY28-2021
FORTRAN IV, GY28-2019
FORTRAN IV Library, GY28-2020
PL/I Compiler, GY28-2051
PL/I Library Computational Subroutines, GY28-2052
Linkage Editor, GY28-2030
System Generation and Maintenance, GY28-2015
Independent Utilities, GY28-2039
On-Line Test Control System, GY28-2042
System Control Blocks, GY28-2011
Time Sharing Support System, GY28-2022
Operator Task and Bulk I/O, GY28-2047

CONTENTS

TIME SHARING SUPPORT SYSTEM

DEFINITIONS

TSSS restricted to system programmers with authority codes 0 or P

RSS–resident support system (suspends TSS/360)
VSS–virtual support system (executes within TSS/360)

Programmer Classifications

MSP–master system programmer, uses RSS; only one MSP connected to system at a time

TSP–task system programmer, uses VSS and, indirectly, RSS; several attached to system at one time, but only one per task

INVOKING TSSS

Action	Result
External interrupt key pressed	MSP connected at operator's terminal, $ written at terminal to invite TSSS input, RSS mode; TSS/360 execution suspended
VSS command at user terminal	TSP connected at same user terminal, $ written at terminal to invite input, VSS mode for task; task execution suspended
CONNECT command issued by MSP followed by RUN or DISCONNECT*	TSP connected at terminal of specified task, $ written at terminal to invite input, VSS mode for specified task; task execution suspended

*CONNECT command, RSS only, can be issued only by MSP

QUALIFICATION

Address Qualification
- real storage addresses–$RM
- virtual storage addresses–$VM
- external storage addresses–implied secondary storage address in operand

Command Qualification
- Indicates private or global (public) AT statements; MSPs and TSPs implant ATs in real, virtual, or shared virtual storage; ATs implanted by RSS (for MSP or TSP) are globally qualified; those implanted by VSS (for TSP) in shared virtual storage can be optionally qualified as global.

Qualification Relationships

Issued by	In storage (type)	Implanted by	Qualification assigned	AT-Table		
				TSP	Global	RSS
TSP	real	RSS	global			X
TSP	virtual	VSS	private	X		
	virtual (shared)	VSS	private global	X	 X	
MSP	real	RSS	global			X
MSP	virtual	RSS	global			X
MSP	virtual (shared)	RSS	global			X

SYNTAX

The following operators and symbols are used with TSSS commands:

Operators

Arithmetic Operators
+ (addition)
– (subtraction)
x (multiplication)
/ (division)

Boolean operators
& (and)
| (or)
¬ (not)

Relational operators
< (less than)
> (greater than)
= (equal to)

Symbols

External symbols–reference specific data fields (in real or virtual storage)

SP symbols–symbolic names, used by system programmers in TSSS as:
- identifications of (temporary) defined data fields
- pseudonyms for TSS/360 data fields or I/O devices

Specified as: character string; one-to-eight alphameric characters, first alphabetic

System symbols–when used in command statement operands, predefined TSSS symbols reference specific data fields or perform certain functions

Formats for system-symbol specification

$RM[(n)] [.]	real memory
$VM[(n)] [.]	virtual memory
$B (sp symbol)	base address, field containing
$P (sp symbol)	pointer
$L (sp symbol)	length
$T (sp symbol)	type {hex=01, char=02, dec=03}
$S (sp symbol)	size
$R (n)	general-purpose registers n=0–15
$C (n)	extended-control registers n=0–15
$E (n)	floating-point machine registers n=0,2,4, or 6
$DHDR	used with SET command; establishes label for output of DUMP command; maximum header length, 80 bytes. SET $DHDR='SAMPLE OF TSSS DUMP'
$TASK (taskid)	used with DUMP command to produce formatted dump of a task's status indicators; if no taskid specified, current task assumed:

Print Line Number	Contents
1	primary header
2	$DHDR
3	taskid, user IDF
4	current PSW
5, 6, 7	general registers
8, 9, 10	control registers
11, 12	floating-point registers
13, 14, 15, 16	old virtual PSWs
17-21	TSI
22-34	TSI header

PSW symbols in RSS

$PSW	current PSW
$PPSW	program interrupt old PSW
$SPSW	supervisor call interrupt old PSW
$XPSW	external interrupt old PSW
$IPSW	I/O interrupt old PSW
$MPSW	machine-check interrupt old PSW

PSW symbols in VSS

$PSW	current VPSW
$PPSW1	recoverable data set paging error VPSW
$PPSW2	program interrupt old VPSW
$SPSW	supervisor call interrupt old VPSW
$XPSW	external interrupt old VPSW
$APSW	asynchronous I/O interrupt old VPSW
$IPSW	I/O interrupt old VPSW
$TPSW	timer interrupt old VPSW
$VPSW	VSS activation old VPSW

For MSP only

$CAW	channel address word
$CSW	channel status word
$TASKID	for MSP identification number of task that was current when RSS received control for TSP identification number of TSPs current task

```
                                 { CSECT,PSECT,ENTRY }
$ID(L'xxxxxxxx') used to obtain  {   POINT (RM)      } name
                                 { CSECT,PSECT (VM)  }
```

whose address is nearest to but not greater than the address specified by 'xxxxxxxx'

$MAP used to obtain dump of TSS/360 supervisor or a specific task

```
    (( C'xxx'       ) [ (                                 ) ] )
    (( X'xxx'       ) [ ( ,,,[number of records],mode set ) ] ) [            ]
$IO (( sp symbol    ) [ ( ,cylinder,track[,record,]       ) ] ) [ .(o,l,t,s) ]
    (( decimal      ) [ (                                 ) ] )
    ((   integer    ) [ (                                 ) ] )
```

Note: $IO, used with SET command, allows user to card-to-tape: SET $IO (X'0182')=$IO(X'000C')

terminal-to-tape: SET$IO(X'0182')=C'AT INTPROC COLLECT COLAREA=$IPSW'

```
     ( { C'xxxx'         }      ) [            ]
$VAM ( { X'xxxx'         } [,y] ) [ .(o,l,t,s) ]
     ( { decimal integer }      ) [            ]
     ( { sp symbol       }      )
```

```
      { X'xxxx'   }
$DOUT { sp symbol }
```

$DOUT = device for DUMP commands

```
$AT[.location]     }
$PATCH [.location] } used with DISPLAY, DUMP, or REMOVE
```

$STATUS used with DUMP command to produce formatted dump of all system-status indicators:

Print Line Number	Contents
1	primary header
2	$DHDR
3	taskid, CPUid
4	current PSW
5, 6, 7	general registers
8, 9, 10	control registers
11, 12	floating-point registers
13, 14, 15	old PSWs
16	channel address word, channel status word
17-21	TSI
22-34	XTSI header

Data Fields

Data fields are defined by:

Symbols–system, external, and SP

Absolute addresses–Hexadecimal storage addresses (in real or virtual storage) referenced using L-notation.

Format: L'xxxxxxxx' Implied Attributes: o=0
l=1 byte
t=hex
s=1 byte

Indirect Addressing–multiple levels of indirect addressing may be specified with % sign

Format: data field %

where data field is generalized representation of all types of symbols and other means of designating storage addresses

Subscripting–array name and subscript used to reference element in array

Format: data field (m)

m=0 will point to first element in array; in general, m=(x-1) will point to x^{th} element in array

Range–an address range

Format: data field_1 : data field_2

Immediate Attribute Designation–To define or change implicit attributes use "immediate attribute designation"

Format: data field (o,l,t,s)

where o = offset
l = length

$$t = \text{type}\left\{\begin{array}{l} I = \text{decimal integer} \\ X = \text{hexadecimal} \\ C = \text{character} \end{array}\right\}$$

s = size

Literal data–immediate data in input stream that becomes content of nonaddressable field

Type	Max Value	Implied Length	Example
Decimal	$2^{31} - 1$	4 bytes	128
Hexadecimal	determined by length in 256-byte input buffer	maximum length of 256 bytes	X'134abc'
Character	not applicable	specified number of characters	C'CHAR''''S'

Constant data–address constants only; value equals storage address of symbol; specified as: A'symbol'

TSSS Command Functions

Command	Function
AT	Designates point in program where AT statement is to be executed
CALL	Initiates execution of prestored set of command statements
COLLECT	Moves data from specified area into specified collection area
CONNECT	Issued by MSP only; TSP is connected to VSS at terminal of specified task
DEFINE	Defines temporary symbols and allocates necessary storage
DISCONNECT	Disconnects TSSS capability from terminal; restores TSS/360 (except for patches); permanently transfers control to TSS/360
DISPLAY	Writes data on terminal
DUMP	Writes data on specified output device
END	Terminates reading of input device used for prestored statement sets
IF	Designates conditional statement; if condition satisfied, statement is executed
PATCH	Alters contents of specified data field; keeps record of patches
QUALIFY	Establishes implicit qualification (real memory, virtual memory, or global) for subsequent operands
REMOVE	Deletes ATs and associated dynamic statements, or deletes patches
RUN	Transfers control to TSS/360; ATs can then be executed
SET	Alters content of specified data field
STOP	Halts TSS/360 or specific task

TSSS COMMANDS

Command	Operands
AT	$\text{address}\left\{\begin{matrix}\text{L.‘xxxxxxxx’}\\ \text{external symbol}\\ \text{sp symbol}\\ \text{system symbol}\end{matrix}\right\}\left[\text{,...}\right]$
CALL	$\left\{\begin{matrix}\text{X‘xxxx’}\\ \text{C‘xxxx’}\\ \text{Sp symbol}\\ \text{decimal integer-device address}\end{matrix}\right\}$
COLLECT	$\text{sp symbol} = \left\{\begin{matrix}\text{data field}\\ \text{literal}\end{matrix}\right\}\left[\text{,...}\right]$
CONNECT (executed by MSP only)	$\text{task I/D number} = \left\{\begin{matrix}\text{sp symbol}\\ \text{\$TSKID}\\ \text{constant}\end{matrix}\right\}$
DEFINE (format 1) (format 2)	sp symbol [.(o,l,t,s)] [,...] $\text{sp symbol=}\left\{\begin{matrix}\text{external symbol}\\ \text{sp symbol}\\ \text{system symbol}\\ \text{address-L‘xxxxxxxx’}\end{matrix}\right\}\left[\text{,(o,l,t,s)}\right]\left[\text{,...}\right]$
DISCONNECT	none
DISPLAY	$\left\{\begin{matrix}\text{data field}\\ \text{literal}\end{matrix}\right\}\left[\text{,...}\right]$
DUMP	$\left\{\begin{matrix}\text{data field}\\ \text{literal}\end{matrix}\right\}\left[\text{,...}\right]$
END	none
IF	expression
PATCH	$\text{data field1} = \left\{\begin{matrix}\text{data field2}\\ \text{literal}\end{matrix}\right\}\left[\text{,...}\right]$
QUALIFY	$\left\{\begin{matrix}\textit{real} & \text{\$RM[(n)]} & \text{[.command operand]}\\ \textit{virtual} & \text{\$VM[(n)]} & \text{[.command operand]}\\ \textit{global} & \text{\$RM[(n)]} & \end{matrix}\right\}$
REMOVE	$\left\{\begin{matrix}\text{\$AT}\\ \text{\$PATCH}\end{matrix}\right\}\left[\left\{\begin{matrix}\text{external symbol}\\ \text{sp symbol}\\ \text{system symbol}\\ \text{L‘xxxxxxxx’}\end{matrix}\right\}\right]\left[\text{,...}\right]$
RUN	$\left[\text{address-}\left\{\begin{matrix}\text{external symbol}\\ \text{sp symbol}\\ \text{system symbol}\\ \text{L‘xxxxxxxx’}\end{matrix}\right\}\right]$
SET	$\text{data field1} = \left\{\begin{matrix}\text{data field2}\\ \text{literal}\end{matrix}\right\}\left[\text{,...}\right]$
STOP	none
VSS	user identification code

EXAMPLES OF TSSS USAGE

Displaying storage areas

1. display any area on direct access device

   ```
                    ( symbolic de- )
   DISPLAY $IO (C' <   vice addr    > ',cylinder, track,
                    ( actual addr  )

                    record). (o,l,t,s)
   ```

   defaults: If (cylinder, track, record) any of these is defaulted, all to right are defaulted; default for offset (o)=0, length (1)=1, type (t)=hex, size (s)=1 byte

2. display byte '3A' on cylinder 0, track=1, record=1:
   DISPLAY $IO(C'12',0,1,1).(X'3A')

3. display a register
   DISPLAY $r(15)

4. display range of real core by use of literals
   DISPLAY L'1AC': L'22C'

5. display range of VM by use of literals
   DISPLAY $VM.L'15004'.(,20) in hex format
   DISPLAY $VM.L'15004'.(,20,C) in character format

6. display range of VM using symbols
   DISPLAY $VM.CZAHAR.(X'166',1)

7. display owner ID of location in core
   user: DISPLAY $ID (L'2B988')
   system: CHBSTE 2B988

Dumping storage areas

1. dump range in core by use of literals
   DUMP L'0':L'7ffff' (assumed dump of real core)
   DUMP L'0':L'7ffffff0' (virtual dump)

2. dump range by use of symbols
   DUMP LABEL1:LABEL2

3. dump range in VM when not certain of qualification
   DUMP $VM.L'0':$VM.L'1000'

4. dump (in hex) VAM public VTOC on cylinder 0
   DUMP $IO(C'12',0) (in hex)

   ```
              ( { C'sda'            }    )
   DUMP $IO  (  {                   } ,0  ) . (,,C)
              ( { X'physical path'  }    )
   ```

   Note: Even with default for length=1(,,C) entire cylinder dumped in this case

5. dump VM and real core, when not certain of qualifications, and get storage maps of both.

   ```
   QUALIFY     $RM
   DUMP        $MAP
   DUMP        L'0':L'7ffff'
   QUALIFY     $VM
   DUMP        $MAP
   DUMP        L'0':L'7ffffff0'
   ```

Modifying storage areas

1. patch an area on a device; set byte '3A' on cylinder=0, track=1, record=1 at SDA=12 equal to X'40'
   ```
   SET $IO (C'12',0,1,1).(X'3a') = X'40'
   ```
2. change VAM disk ID on SDA=22
   ```
   SET$IO(C'22',0,0,3).(8,6) = C'NARESC'
   ```
3. patch area in user code, VM (e.g., restbl header for specific taskid)
   ```
   SET $VM(X'0014').L'90AFBO' = X'C3'
   ```

Assign tape device for dumps

```
$SET DOUT = $IO (C'43')
```

Indirect addressing

1. display registers in XTSI of the current task (when qualified for $RM)
   ```
   DISPLAY L'188'%.(X'C')%.(X'50',64)
   ```

2. display contents of area pointed to by register 13, offset by 20 (when qualified for $RM)
   ```
   DISPLAY $R(13)%.(X'20',20)
   ```

CONTENTS

SYSTEM MACRO INSTRUCTIONS

This section contains the system programmer oriented macro instructions. It contains alphabetic listings of the macro instructions with their definitions and formats.

MACRO INSTRUCTION DEFINITIONS

ACCTSUBR–accounting subroutine call
ADDEV–add device to task symbolic device list
ADDPG–add virtual storage pages
ADSPG–add shared virtual storage pages
ATCS–activate terminal communication subprocessor
ATPOL–poll for pending attention interrupt
ATTACH–attach a task to the system
AUXSET–create overload/overdraw interrupt control blocks
AVAUX–available auxiliary remaining count
AWAIT–wait for interrupt
CANCL–cancel real-time interruption
CHANGE–change schedule table entry
CHECK–wait for, test for completion of read or write
CKALOC–check for terminal MTT status
CKCLS–check protection class
CLEARQ[1]–clear terminal device status
CLIC–read command from SYSIN
CLIP–read command from SYSIN
CLOSE (MSAM)–disconnect data set from user's problem program
CLOSE (TAM)–remove communication lines from use
CNSEG–connect segment to shared page table
CONN–connect a multiterminal task
CRTSI–create task status index
DCB (MSAM)–set up data control block
DCB (TAM)–set up data control block
DCBD–specify DCB DSECT
DCLASS–specify privilege class
DCON–disconnect a multiterminal task
DDEF–define data set
DELET–enter delete program
DELPG–delete virtual storage pages
DFTRMENT–define polling list
DLINK–transfer to dynamic loader for external symbol resolution
DLTSI–delete task status index
DSSEG–disconnect shared page table from segment
ENTER–enter privileged service routine
ERROR–indicate supervisor-detected error
FINDDS–locate JFCB corresponding to data set name
FINDJFCB–locate JFCB and ensure volume mounting
FINDQ[1]–find terminal requiring work
FINISH (MSAM)–end of data set
FREEQ[1]–drop a terminal device
GET (MSAM)–get record
HASH–develop hash value for symbol
INVOKE–transfer control
IOCAL–I/O call
ITI–inhibit task interrupts
LCD–indicate line code
LIBESRCH–locate program module in external library
LSCHP–list changed virtual storage pages
LVPSW–load virtual program status word
MOVXP–move page table entries
OPEN (MSAM)–prepare data control block for processing
OPEN (TAM)–prepare DCB for processing

[1] for use only in MTT application programs

PSCVC–enter program control system
PGOUT–write virtual storage pages to external storage
PRESENT–present current schedule level
PTI–permit task interrupts
PULSE–pulse schedule table entry level
PURGE–purge I/O operations
PUT (MSAM)–put record
RCR–resource control instruction
RCR OPEN
RCR CLOSE
RCR UPDATE
RCR RATION
RCR VACATE
RCR LOGOFF
RDI–reset drum interlock
READ (TAM)–read from another terminal
READQ[1]–initiate read operation to terminal
REDTIM–read elapsed real time
RESET–reset device supression flag
RESUME–return to calling program
RJELC–remote job entry line control
RMDEV–remove device from task symbolic device list
RSPRV–restore privilege
RSTTIM–reset system time
RTRN–enter command analyzer to end run
SAMPLE–sample system status table
SCHED–schedule table entry
SCRTSI–special create-task status index
SETAE–set asynchronous entry
SETSYS–set system table field
SETTIMER–set real-time intervals from resident program
SETTOD–set time of day
SETTR–set real time interval
SETTU–set user timer
SETUP–set up task status index field
SETUR–unit record device set up
SETYMD–set year, month, and day
SETXP–set external page table entries
SETXTS–set up extended task status index field
SPATH–set I/O device path
STORE–store register contents
SYSER–indicate nonresident-program-detected error
TSEND–force time slice end
TWAIT–wait for terminal I/O interrupt
UPDTUSER–update user tables
USAGE–display resource usage statistics
VSEND–send message to another task
VSENDR–send message to task and await response
WAIT[2]–wait for terminal stimuli
WRITE (TAM)–write message
WRITEQ[1]–write message to terminal
XTRCT–extract task status index field
XTRSYS–extract system table field
XTRTM–extract accumulated CPU time
XTRXTS–extract extended task status index field
ZEROSST–zero system status table

Inner Macro Instructions

CHDERMAC–generate error message
CHDINNRA–general type-1 or type-2 linkage
CHDPSECT–reserve storage for parameter list

[1] for use only in MTT application programs
[2] can be used in both TSS mode and MTT mode

MACRO INSTRUCTION FORMATS

Name	Operation	Operands
[symbol]	ACCTSUBR	none
[symbol]	ADDEV	$\left[\text{device number}\begin{Bmatrix}\text{value}\\ (\underline{0})\end{Bmatrix}\right]$
		return data: if reg 0 high-order bit = 1; exceeds allowable device limit
[symbol]	ADDPG	$\left[\text{page count}\begin{Bmatrix}\text{value}\\ (\underline{1})\end{Bmatrix}\right]$ $\left[\begin{Bmatrix}\text{,start address -addrx ,half page protection}\\ \text{class code}\\ (\underline{0})\end{Bmatrix}\right]$
		register notation value code — value — definition A — 1 — both halfpages nonprivileged read/write AB — 4 — first halfpage nonprivileged read/write, second halfpage nonprivileged read-only AC — 7 — first halfpage nonprivileged read, second halfpage privileged BA — 2 — first halfpage nonprivileged read-only, second halfpage privileged B — 5 — both halfpages nonprivileged read-only BC — 8 — first halfpage nonprivileged read-only, second halfpage privileged CA — 3 — first halfpage privileged, second halfpage nonprivileged read/write CB — 6 — first halfpage privileged, second halfpage nonprivileged read-only C — 9 — both halfpages privileged
[symbol]	ADSPG	$\left[\text{start}\begin{Bmatrix}\text{addrx}\\ (\underline{1})\end{Bmatrix}\right]\left[\text{,page count}\begin{Bmatrix}\text{value}\\ (\underline{0})\end{Bmatrix}\right]$ $\left[,\begin{Bmatrix}\text{shared page table number value,}\\ \text{protection class code}\\ (15)\end{Bmatrix}\right]$
		see ADDPG for codes
[symbol]	ATCS	none
		note: preset registers 0 and 1 with reg 0 — either the virtual storage address of TCT slot containing processing options or all Fs to denote FREEQ ALL reg 1 — when TCT slot indicates message to be sent with freeing option, should be set with address of message length followed by text; otherwise set to 0
[symbol]	ATPOL	program address-addrx [,switch-addx]
	ATTACH	none
		return data: reg 1 set to code — meaning 0 — no TSI match found xxxxxx — virtual storage address of system TCT slot
[symbol]	AUXSET	none
[symbol]	AVAUX	[amount-value,] location-addr

Name	Operation	Operands
[symbol]	AWAIT	none
[symbol]	CANCL	[task status index address-(reg)] $,\text{interruption rtne adcon}\begin{Bmatrix}\text{symbol}\\ \text{(reg)}\end{Bmatrix}$ return data: reg 15 set to X'15' for normal returns
[symbol]	CHANGE	$\left[\text{level}\begin{Bmatrix}\text{addr}\\ (\underline{15})\end{Bmatrix}\right]$
	CHECK	(see TAM)
[symbol]	CKALOC	$\left[\text{devnbr}\begin{Bmatrix}\text{comp-value}\\ \text{dev-value}\\ (0)\end{Bmatrix}\right]$ return data: reg 0 is set to code meaning 0 a user currently has control of the terminal 1 was in control of RTAM; now under user control 2 terminal is associated with MTT; control of I/O is not given to user 3 user control relinquished successfully
[symbol]	CKCLS	$\left[\text{start address}\begin{Bmatrix}\text{addrx}\\ (\underline{1})\end{Bmatrix}\right]\left[,\text{halfpage count}\begin{Bmatrix}\text{value}\\ (\underline{0})\end{Bmatrix}\right]$ $\left[,\text{segment length}\begin{Bmatrix}\text{VAR}\\ \underline{\text{nonvariable}}\end{Bmatrix}\right]$ return data: reg 0 low-order byte set to code protection class 0 page unassigned 1 user read/write (least restrictive) 3 user read only 7 user cannot read or write (most restrictive)
[symbol]	CLEARQ	relative line number-value return data: code meaning 00 normal return 04 invalid relative line number 08 busy 0C attention interruption received; normal clearing functions performed
[symbol]	CLIC	none
[symbol]	CLIP	none
	CLOSE	(see MSAM and TAM)
[symbol]	CNSEG	$\left[\begin{Bmatrix}\text{segment number-value, shared page}\\ \text{table number-value}\\ (\underline{1})\end{Bmatrix}\right]$
[symbol]	CONN	none note: registers 0–5 must be preset with the indicated values reg parameter 0,1 eight-character application program name in EBCDIC; if less than eight characters, pad with trailing blanks 2 virtual storage address of first TCT slot 3 virtual storage address of the first buffer page

Name	Operation	Operands
		4 left half-word contains maximum number of lines allowed simultaneous connection; right half-word contains buffer length 5 left half-word is set to number of pages allocated for the TCT and buffer pages return data: low-order half-word of register 0 is set to X'FFFF' if task already a multiterminal task
[symbol]	CRTSI	none return data: reg 1 set to task ID or to 0's if TSI limit exceeded
	DCB	(see MSAM or TAM)
	DCLASS	$\left[\text{class}\left\{\begin{matrix}\underline{\text{USER}}\\ \text{PRIVILEGED}\end{matrix}\right\}\right]$
[symbol]	DCON	none
[symbol]	DELET	none
[symbol]	DELPG	$\left[\text{start address}\left\{\begin{matrix}\text{addrx}\\ (\underline{0})\end{matrix}\right\}\right]\left[\text{,page count}\left\{\begin{matrix}\text{value}\\ (\underline{1})\end{matrix}\right\}\right]$
	DFTRMENT	(see TAM)
[symbol]	DLINK	none
[symbol]	DLTSI	none
[symbol]	DSSEG	$\left[\text{shared page table number}\left\{\begin{matrix}\text{value}\\ (\underline{1})\end{matrix}\right\}\right]$
[symbol]	ENTER	none (reg. 15 must be preloaded with enter code; see table of enter codes, following in this section.)
[symbol]	ERROR	errtype-code, compat-integer, opt-integer, idno-integer code definition 1 minor software error 2 major software error 3 hardware failure
[symbol]	FINDDS	dsname-addr, byte-addr, area-addr return data: reg. 15 = code (hex) definition 00 JFCB found or created 04 no JFCB found; no request to create one 08 no JFCB found; request to create one, but no dsname in catalog 0C no JFCB found; DDEF could not create one; space unavailable
[symbol]	FINDJFCB	ddname-addr,byte-addr,area-addr
[symbol]	FINDQ	[relative line number-value] return data: code meaning 00 no work 04 invalid relative line number 08 initial connection of device 0C attention received from terminal device 10 solid I/O error on terminal line 14 message out, complete 18 message in, complete

Name	Operation	Operands
	FINISH	(see MSAM)
[symbol]	FREEQ	$\begin{Bmatrix}\text{relative line number-value}\\ \text{ALL}\end{Bmatrix}$ [,message pointer-addr] $\left[,\text{ disconnect}\begin{Bmatrix}\text{PD}\\ \underline{\text{LD}}\end{Bmatrix}\right]$
		return data: reg 15 = <u>code</u> <u>meaning</u> 00 normal return 04 invalid relative line number
	GET	(see MSAM)
[symbol]	HASH	name,value
[symbol]	INVOKE	address program adcon-addrx
[symbol]	ITI	none
[symbol]	LCD	SDA = symbolic device address
		return data: low-order byte of register 0 is set to one of the following codes: <u>code</u> <u>meaning</u> 00 no entry for specified SDA 01 1050 PTTC/8 (folded) 02 2741 Correspondence (folded) 03 2741 PTTC/8 (folded) 04 teletypewriter ASCII (folded) 05 1052-7 EBCDIC
[symbol]	LIBESRCH	list-addr,not found exit-addr
		return data: 9 words of information loaded into address at list-addr+3
[symbol]	LSCHP	$\left[\text{start}\begin{Bmatrix}\text{addrx}\\ (\underline{1})\end{Bmatrix}\right]\left[\text{,page count}\begin{Bmatrix}\text{value}\\ (\underline{0})\end{Bmatrix}\right]$
		return data: page condition code for page 'n' is set in bits 2n-2 and 2n-1 in reg 0; bit <u>pair</u> <u>definition</u> 00 page in core changed 01 page in core unchanged 10 page not in core changed 11 page not in core unchanged
[symbol]	LVPSW	$\left[\text{PSW address}\begin{Bmatrix}\text{addrx}\\ (\underline{1})\end{Bmatrix}\right]$
[symbol]	MOVXP	$\left[\text{start address}\begin{Bmatrix}\text{addrx}\\ (\underline{0})\end{Bmatrix}\right]\left[\text{,to address}\begin{Bmatrix}\text{addrx}\\ (\underline{1})\end{Bmatrix}\right]$ $\left[\text{page count}\begin{Bmatrix}\text{value}\\ (15)\end{Bmatrix}\right]$

MSAM (multiple sequential access method)

Name	Operation	Operands
[symbol]	CLOSE	(dcb addr,...)
[symbol]	DCB	$\text{DSORG = MS}\left[\text{,MACRF=}\left\{\begin{matrix}\text{G}\\ \text{P}\end{matrix}\right\}\right]\left[\text{,DDNAME = alphname}\right]$ [,DEVD=code] [,RECFM=code] [,LRECL=absexp] $\left[\text{,RETRY=}\left\{\begin{matrix}\text{N}\\ \text{U}\end{matrix}\right\}\right]\left[\text{,COMBINE=}\left\{\begin{matrix}\text{Y}\\ \underline{\text{N}}\end{matrix}\right\}\right]\left[\text{,POCKET}\left\{\begin{matrix}\text{1}\\ \text{2}\\ \text{ORG}\end{matrix}\right\}\right]$ $\left[\text{,FORMTYPE=}\left\{\begin{matrix}\text{F}\\ \text{S}\\ \text{D}\end{matrix}\right\}\right]\left[\text{,INHMSG=}\left\{\begin{matrix}\text{Y}\\ \underline{\text{N}}\end{matrix}\right\}\right]$ $\left[\text{,MODE=}\left\{\begin{matrix}\text{C}\\ \underline{\text{E}}\end{matrix}\right\}\right]\left[\text{,STACK=}\left\{\begin{matrix}\underline{\text{1}}\\ \text{2}\\ \text{3}\end{matrix}\right\}\right]$ $\left[\text{,PRTSP=}\left\{\begin{matrix}\text{0}\\ \underline{\text{1}}\\ \text{2}\\ \text{3}\end{matrix}\right\}\right]$
	DDEF	$\left\{\begin{matrix}\text{ddname-symbol}\\ \text{PCSOUT}\end{matrix}\right\}\text{,DSORG=PS}$ $\text{,DSNAME=data set name,UNIT=}\left\{\begin{matrix}\text{sda}\\ \text{PC}\\ \text{PR}\\ \text{RD}\end{matrix}\right\}$ Note: For additional optional parameters, see Quick Guide for Users
[symbol]	FINISH	$\text{dcb}\left\{\begin{matrix}\text{addrx}\\ \text{(1)}\end{matrix}\right\}$ return data: reg 15 = code definition 0 normal completion 4 I/O not completed; reissue FINISH until different return code received 8 complete with I/O error
[symbol]	GET	$\text{dcb}\left\{\begin{matrix}\text{addr}\\ \text{(1)}\end{matrix}\right\}\left[\text{,area}\left\{\begin{matrix}\text{addrx}\\ \text{(0)}\end{matrix}\right\}\right]$ return data: reg 15 = code definition 0 normal completion 4 I/O not complete; reissue GET 8 unrecoverable I/O error occurred 12 end of file; reissue FINISH 16 control card sensed
[symbol]	OPEN	[}(dcb-addr,[(opt-code)],{ ...)]
[symbol]	PUT	$\text{dcb}\left\{\begin{matrix}\text{addrx}\\ \text{(1)}\end{matrix}\right\}\left[\text{,area}\left\{\begin{matrix}\text{addrx}\\ \text{(0)}\end{matrix}\right\}\right]$ return data: reg 15 = code definition 0 normal completion 4 I/O not complete, reissue PUT 8 unrecoverable I/O error occurred

Name	Operation	Operands
	MSAM	(Continued)
[symbol]	SETUR	$\text{dcb}\left\{\begin{matrix}\text{addr}\\(1)\end{matrix}\right\}, \text{param}\left\{\begin{matrix}\text{addr}\\(0)\end{matrix}\right\}$ return data: reg 15 = <u>code</u> <u>definition</u> 0 normal completion 4 operation not completed; reissue SETUR 8 unrecoverable error occurred 12 parameter is invalid SYSURS key 16 invalid buffer load key in SYSUCS

[symbol]	PCSVC	none
[symbol]	PGOUT	none return data: reg 0 = four-bit group codes describing action of each page; bits 0-3 first page, bits 4-7 second, etc. <u>value</u> <u>definition</u> 0000 no error-page transmitted 0011 virtual storage page not assigned to task 0100 request for zero pages 0101 symbolic device not assigned to task 0110 page in, bad device, volume is movable 0111 page in, bad device, volume fixed 1000 page in, medium failure 1001 page out, bad device, volume is movable 1010 page out, bad device, volume fixed 1011 page out, medium failure
[symbol]	PRESENT	none
[symbol]	PTI	none
[symbol]	PULSE	none
[symbol]	PURGE	$\left[\left\{\begin{matrix}\text{action-code, symbolic device number}\\(0)\end{matrix}\right\}\right]$ $\left[\left\{\begin{matrix}\text{,task code}\left\{\begin{matrix}\text{AT}\\\text{ST}\end{matrix}\right\}\text{[,taskid value]}\\(1)\end{matrix}\right\}\right]$ <u>action codes are:</u> AR purge all devices immediately AS purge all devices, let active ones quiesce AL purge all I/O requests immediately, leave TSDL alone AD remove the TSDL SR purge specified device after quiesce <u>task codes mean:</u> AT purge all tasks ST purge specified task only
	PUT	(see MSAM)
[symbol]	RCR	$\text{OPEN, failure-addr}\left[\text{,userid}\left\{\begin{matrix}\text{addr}\\\underline{\text{TCMUID}}\end{matrix}\right\}\right]$ $\left[\text{,user entry}\left\{\begin{matrix}\text{addr}\\\underline{\text{TCMVLU}}\end{matrix}\right\}\right]\left[\text{,AUL entry}\left\{\begin{matrix}\text{addr}\\\text{TCMAUL}\end{matrix}\right\}\right]$ $\left[\text{,logon}\left\{\begin{matrix}\underline{\text{L}}\\\text{U}\end{matrix}\right\}\right]$ $\text{CLOSE, failure-addr}\left[\text{,AUL entry}\left\{\begin{matrix}\text{addr}\\\underline{\text{TCMAUD}}\end{matrix}\right\}\right]$

Name	Operation	Operands
		UPDATE $\left[\text{,user entry}\left\{\begin{array}{l}\text{addr}\\ \underline{\text{TCMVLU}}\end{array}\right\}\right]$
		RATION, failure-addr, type - $\left\{\begin{array}{l}\text{CPU - CPU time (used with RATION only)}\\ \text{CONN - terminal connect time}\\ \text{TASK - a TSS background task}\\ \text{STOR - permanent or temporary}\\ \quad\quad\text{external storage}\\ \text{DA - direct access device}\\ \text{MT - magnetic tape drive}\\ \text{PTR - high speed printer}\\ \text{RPU - reader/punch}\\ \text{BI - BULKIO records read in}\\ \text{BO - BULKIO records written out}\end{array}\right\}$ [,amount-value]
		VACATE, type-code [,amount-value] $\left[\text{,user entry}\left\{\begin{array}{l}\text{addr}\\ \text{TCMVLU}\end{array}\right\}\right]$ [,failure-addr]
		Note: Codes are indicated under RATION.
		LOGOFF, failure-addr
[symbol]	RDI	none
	READ	(see TAM)
[symbol]	READQ	relative line number-value $\left[\text{,TRNSL =}\left\{\begin{array}{l}\text{N}\\ \underline{\text{Y}}\end{array}\right\}\right]$ $\left[\text{,INTRPT =}\left\{\begin{array}{l}\text{Y}\\ \underline{\text{N}}\end{array}\right\}\right]$ $\left[\text{, COMPSEL=value}\right]$
		return data: code meaning 00 normal return 04 invalid relative line number 08 busy 0C attention interruption received from terminal 10 solid error occurred during initiation of starting I/O
[symbol]	REDTIM	none
		return data: regs 0 & 1 = double-precision, fixed-point system time in microseconds
[symbol]	RESET	$\left[\text{device number}\left\{\begin{array}{c}\text{value}\\ \text{'ALL'}\\ (\underline{0})\end{array}\right\}\right]$
[symbol]	RESUME	[area-addrx,(regl-integer[,reg2-integer])] [,RC=integer]

Name	Operation	Operands
[symbol]	RJELC	none
		note: registers 0 and 1 must be preset with the values indicated reg 0 – two byte hexadecimal number representing symbolic device address assigned during system generation reg 1 = code meaning 0 prime the line 1 enable the line 2 disable the line return data: registers 1 and 0 contain return data reg 1 = code meaning 0 SIO successful 4 SIO failed; also examine reg 0 8 path unavailable or invalid input 12 path busy when return code of 4 in reg 1 reg 0 = bits indication 0–1 SIO failure indication (not meaningful to macro instruction execution) 2–3 TIO condition code 4–5 TCH condition code 6–7 SIO condition code 8–23 CSW status byte (if SIO cc=1) 24–31 flags 25 if on means control unit busy 26 if on means an expected interruption was taken by another CPU
[symbol]	RMDEV	$\left[\text{device number} \begin{Bmatrix} \text{dev-value} \\ \text{comp-value} \\ (\underline{0}) \end{Bmatrix} \right]$
		return data: reg 0 high-order bit=1 if symbolic device number not found
[symbol]	RSPRV	none
[symbol]	RSTTIM	none
[symbol]	RTRN	none
[symbol]	SAMPLE	none
[symbol]	SCHED	none
[symbol]	SCRTSI	none
		return data: reg 1 = task I/D
[symbol]	SETAE	$\left[\text{device address} \begin{Bmatrix} \text{value} \\ (\underline{1}) \end{Bmatrix} \right] \left[\text{,task} \begin{Bmatrix} \text{value} \\ (\underline{0}) \end{Bmatrix} \right]$
[symbol]	SETSYS	$\left[\text{field} \begin{Bmatrix} \text{code} \\ (15) \end{Bmatrix} \right]$
		code register notation value implied length definition TOD 1 8 set time of day YMD 2 8 set years, months, days TASKINIT 3 1 set task initiation status field

<table>
<tr><th>Name</th><th>Operation</th><th>Operands</th></tr>
<tr><td>[symbol]</td><td>SETTIMER</td><td>time$\left\{\begin{matrix}\text{integer}\\ \text{(reg [,reg])}\end{matrix}\right\}$, [task status index-(reg)]
, routine$\left\{\begin{matrix}\text{symbol}\\ \text{(reg)}\end{matrix}\right\}$

return data: register 15 is set to
<table>
<tr><th>code</th><th>meaning</th></tr>
<tr><td>04</td><td>system limit reached; no more interruptions may be set up</td></tr>
<tr><td>08</td><td>time specified is not in the future</td></tr>
<tr><td>0C</td><td>normal return</td></tr>
</table></td></tr>
<tr><td>[symbol]</td><td>SETTOD</td><td>none (preload regs 0 & 1 with time of day)</td></tr>
<tr><td>[symbol]</td><td>SETTR</td><td>none (preload regs 0 & 1 with time limit)
return data:
reg 15 = X‘10’ if system limit for real-time interrupts is reached</td></tr>
<tr><td>[symbol]</td><td>SETTU</td><td>$\left[\text{time}\left\{\begin{matrix}\text{value}\\ (\underline{1})\end{matrix}\right\}\right]$</td></tr>
<tr><td>[symbol]</td><td>SETUP</td><td>$\left[\text{field}\left\{\begin{matrix}\text{code}\\ (\underline{15})\end{matrix}\right\}\text{,content register}\left\{\begin{matrix}\text{odd-value}\\ (\underline{1})\end{matrix}\right\}\right]$
<table>
<tr><th>code</th><th>register notation value</th><th>implied length</th><th>definition</th></tr>
<tr><td>USERID</td><td>1</td><td>8</td><td>set user ID field</td></tr>
<tr><td>SYSIN</td><td>3</td><td>2</td><td>set input data set location</td></tr>
<tr><td>SYSOUT</td><td>4</td><td>2</td><td>set output data set location</td></tr>
<tr><td>BSN</td><td>5</td><td>1</td><td>set batch sequence number</td></tr>
<tr><td>CONV</td><td>10</td><td>1</td><td>set conversational task flag</td></tr>
<tr><td>ITMFLG</td><td>12</td><td>1</td><td>set intertask message flag</td></tr>
<tr><td>XPR</td><td>13</td><td>2</td><td>set external priority flag</td></tr>
<tr><td>AUTH</td><td>14</td><td>1</td><td>set privilege</td></tr>
<tr><td>STE</td><td>15</td><td>1</td><td>set schedule table entry field</td></tr>
<tr><td>MAV</td><td>16</td><td>2</td><td>set maximum auxiliary storage field</td></tr>
</table></td></tr>
<tr><td>[symbol]</td><td>SETXP</td><td>none</td></tr>
<tr><td>[symbol]</td><td>SETXTS</td><td>$\left[\text{field}\left\{\begin{matrix}\text{SET 24}\\ \text{ESTIM}\\ (\underline{15})\end{matrix}\right\}\right]$</td></tr>
<tr><td>[symbol]</td><td>SETYMD</td><td>none (preload regs 0 and 1 with year, month and day)</td></tr>
</table>

Name	Operation	Operands
[symbol]	SPATH	[flag { code { POF-set units partitioned flag off (0) PON-set units partitioned flag on (1) SOF-set units malfunction flag off (1) SON-set units malfunction flag on (1) } (0) }] [{ ,component-code, device address-hexinteger (1) }]
		component codes: 1 - I/O device only 2 - control unit only 3 - control unit and I/O device 4 - channel unit only 5 - channel and I/O device 6 - channel and control unit 7 - channel, control unit, and I/O device
[symbol]	STORE	area-addrx,(regl-integer[,reg2-integer])
[symbol]	SYSER	errtype-integer,dump-integer,optl-integer, opt2-integer,opt3-integer,idno-integer

TAM (terminal access method)

[symbol]	CHECK	decb { addrx (1) }
[symbol]	CLOSE	(dcb-addr,...) [MF= { (E,list- { addrx (1) }) }]
[symbol]	DCB	[DDNAME=symbol] [,DSORG=CX] [,MACRF= { (R) (W) (R,W) }] [,BUFNO=absexp] [,BUFL=absexp] [,BFTEK=D] [,EXLST=relexp] [,SYNAD=relexp]
[symbol]	DDEF	ddname-symbol,DSORG=CX, DSNAME=data set name, UNIT=sda Note: For additional optional parameters, see Quick Guide for Users
[symbol]	DFTRMENT	[DIAL=(integer,...)] [,ADRID=(adrid-characters,...)] [,POLLID=(pollid-characters,...)]
[symbol]	OPEN	({dcb-addr,} ......) [MF= { (E,list- { addrx (1) }) }]
[symbol]	READ	decb { symbol (1) } ,type-code,dcb-addr, area { addr 'S' } ,length { value 'S' 'C' } [argl-addr] ,[arg2-code] [,MF= { L E }]
[symbol]	WRITE	decb { symbol (1) } ,type-code,dcb-addr, [area-addr] ,length-value, [arg1-addr] ,[arg2-code] [,MF= { L E }]

Name	Operation	Operands
[symbol]	TSEND	none
[symbol]	TWAIT	none
[symbol]	UPDTUSER	none
[symbol]	USAGE	area-addr [,userid-addr]
[symbol]	VSEND	none
		return data: reg 15 = code definition 00 recipient task cannot be found 04 message not acceptable 08 message sent
[symbol]	VSENDR	msg-text,reply-addr,reply-length, message code-value, sending taskid-addr
symbol	WAIT	$\left[\text{environment} \begin{Bmatrix}\text{A-multitermenal}\\ \underline{\text{T}}\text{-standard TSS}\end{Bmatrix}\right]$
		note: when standard TSS environment specified, register 0 must be set with a pointer to the system TCT slot for that task
	WRITE	(see TAM)
[symbol]	WRITEQ	relative line number-value ,area-addr, length-value $\left[,\text{INTRPT}=\begin{Bmatrix}\text{Y}\\ \underline{\text{N}}\end{Bmatrix}\right] \left[,\text{BREAK}=\begin{Bmatrix}\text{Y}\\ \underline{\text{N}}\end{Bmatrix}\right]$ $\left[,\text{COMPOUT=value}\right] \left[,\text{TRNSOUT} = \begin{Bmatrix}\text{Y}\\ \underline{\text{N}}\end{Bmatrix}\right]$ $\left[,\text{RESP}=\begin{Bmatrix}\text{Y}\\ \underline{\text{N}}\end{Bmatrix}\right] \left[,\text{COMPIN=value}\right]$ $\left[,\text{TRNSIN}=\begin{Bmatrix}\text{N}\\ \underline{\text{Y}}\end{Bmatrix}\right]$
		return data: code meaning 00 normal return 04 invalid relative line number 08 busy 0C attention interruption received from terminal 10 solid error during start-I/O 14 message length exceeds 4080 bytes

Name	Operation	Operands			
[symbol]	XTRCT	$\left[\text{field}\begin{Bmatrix}\text{code}\\(\underline{15})\end{Bmatrix}\right]$			
		code	register notation value	implied length	definition
		USERID	1	8	extract user ID field
		PRIORITY	2	1	extract priority fld
		SYSIN	3	2	extract sysin symbolic dev addr
		SYSOUT	4	2	extract sysout sym dev addr
		BSN	5	1	extract batch sequence number
		SOPRIV	6	1	operator privilege
		SPPRIVF	7	1	system programmer, nonprivileged
		SRPIV	8	1	system programmer, privileged
		UPRIV	9	1	user privilege
		CONV	10	1	conversational task flag
		TASKID	11	2	task ID field
		XPR	12	1	external priority flag
		ITMFLG	13	2	intertask message flag
		AUTH	14	1	extract privilege field
		PENDIO	15	1	pending I/O operations count field
		MAV	16	2	auxiliary storage requirement field
		DISK	17	2	auxiliary storage count field
		return data: regs 0 and 1=extracted TSI field; right justified; number of bytes equals implied length			
	XTRSYS	$\left[\text{field}\begin{Bmatrix}\text{code}\\(\underline{15})\end{Bmatrix}\right]$			
		code	register notation value	implied length	definition
		TOD	1	8	get time of day
		YMD	2	8	get year,month,days
		TASKINIT	3	1	get task initiation status
		AVAUX	5	4	get available auxiliary count field
		return data: regs 0 and 1=extracted TSI field; right justified; number of bytes equals implied length			
[symbol]	XTRTM	none return data: reg 1=total accumulated task CPU time			

<table>
<tr><th>Name</th><th>Operation</th><th>Operands</th></tr>
<tr><td>[symbol]</td><td>XTRXTS</td><td>[field $\left\{\begin{matrix}\text{code}\\ (15)\end{matrix}\right\}$]
<table>
<tr><th>code</th><th>register notation value</th><th>implied length</th><th>definition</th></tr>
<tr><td>UTIME</td><td>1</td><td>4</td><td>user-time field</td></tr>
<tr><td>ATIME</td><td>2</td><td>4</td><td>accumulated-time field</td></tr>
<tr><td>ESTIM</td><td>3</td><td>4</td><td>estimated run-time field</td></tr>
<tr><td>TWAIT</td><td>4</td><td>2</td><td>number of TWAIT field</td></tr>
<tr><td>AWAIT</td><td>5</td><td>4</td><td>number of AWAIT field</td></tr>
<tr><td>TSLICE</td><td>6</td><td>4</td><td>number of time slices field</td></tr>
<tr><td>AUX-IN</td><td>7</td><td>4</td><td>number of page-ins from aux storage</td></tr>
<tr><td>EXT-IN</td><td>8</td><td>4</td><td>number of page-ins from external storage</td></tr>
<tr><td>AUX-OUT</td><td>9</td><td>4</td><td>number of page-outs to aux storage</td></tr>
<tr><td>EXT-OUT</td><td>10</td><td>4</td><td>number of page-outs to external storage</td></tr>
<tr><td>MDISK</td><td>11</td><td>2</td><td>maximum pages used on aux disk field</td></tr>
</table>
return data: regs 0 and 1 = extracted field from XTSI; right justified, padded 0's; number of bytes = implied length</td></tr>
<tr><td>[symbol]</td><td>ZEROSST</td><td>none</td></tr>
</table>

INNER MACRO INSTRUCTIONS

	CHDERMAC	mesno-integer,[opnm-characters], [opva-characters],[opvb-characters], [opvc-characters][,S=integer]
symbol	CHDINNRA	[paraone-$\left\{\begin{matrix}\text{addrx}\\ (1)\end{matrix}\right\}$],[parazero]$\left\{\begin{matrix}\text{addrx}\\ (0)\end{matrix}\right\}$] [([sublista-$\left\{\begin{matrix}\text{symbol}\\ \text{integer}\end{matrix}\right\}$][,sublistb-integer])], [entrcd-absexp,][mcrcd-code]
[symbol]	CHDPSECT	[loc-addr],[align-$\left\{\begin{matrix}\text{0F}\\ \text{0H}\end{matrix}\right\}$] [,string-text]

SYSTEM ENTER CODE TABLE

Decimal	Hex	Name	Entry point	PSECT
		TAM		
0	00	READ/WRITE	CZCYM1	CZCYMP
1	1	BATCH MONITOR	CZABAE	CZABAE
		INTERRUPT HANDLING		
16	10	SIR	CZCJSA	CZCJSP
17	11	DIR	CZCJDA	CZCJDP
18	12	INTINQ	CZCJIA	CZCJIP
19	13	STIMER/TTIMER	CZCJA1	SYSJAR
30	1E	Test Case 1	TEST1V	TEST1R
31	1F	Test Case 2	TEST2V	TEST2R
		SAM		
32	20	READ/WRITE	CZCRAS	CZCRAP
33	21	CHECK	CZCRCS	CZCRCP
34	22	CNTRL	CZCRBS	CZCRBP
36	24	POINT	CZCRMA	CZCRMP
37	25	BSP	CZCRGA	CZCRGP
		VM ALLOCATION		
48	30	GETMAIN (R)	CZCH2	CZCH5
49	31	GETMAIN (PAGE)	CZCG2	CZCG5
50	32	FREEMAIN (R)	CZCH3	CZCH5
51	33	FREEMAIN (PAGE)	CZCG3	CZCG5
		VAM GENERAL SERVICES		
56	38	VDMEP	CZCQK1	CZCAKP
57	39	DUPOPEN	CZCQK1	CZCQKP
58	3A	DUPCLOSE	CZCEY1	CZCEYP
		VISAM		
61	3D	VISAM SETL	CZCPC3	CZCPC3
		VAM		
62	3E	VSAM PUT	CZCOS3	CZCOS3
63	3F	LIBESRCH	CZCDL3	CZCDLP
64	40	READ/WRITE	CZCPE1	CZCPEP
65	41	ESETL	CZCPD1	CZCPIP
66	42	RELEX	CZCPG1	CZCPIP
67	43	DELREC	CZCPH1	CZCPEP
68	44	FIND	CZCOJ1	CZCOJP
69	45	STOW	CZCOK1	CZCOKP
70	46	ADE	CZCPL1	CZCPLP
71	47	GETPAGE	CZCPI1	CZCPIP
72	48	INSPACE	CZCOD1	CZCODP
73	49	DELPAGE	CZCOD2	CZCODP
74	4A	VSAM PUT EXTERNAL USER	CZCOS1	CZCOSP
75	4B	VSAM PUT INTERNAL	CZCOS2	CZCOSP
76	4C	MOVEPAGE	CZCOC1	CZCOCP
77	4D	FLUSHBUF	CZCOV1	CZCOVP
78	4E	VISAM GET PAGE INPUT	CZCPI2	CZCPIP
79	4F	VISAM GET PAGE OUTPUT	CZCPI3	CZCPIP
		MACRO COMMAND LANGUAGE		
80	50	GATRD/GATWR	CZATC2	CZATCP
81	51	WTO	CZABQ1	CZABQR
82	52	WTOR	CZABQ1	CZABQR
83	53	ERASE	CZAEJ7	CZAEJR
84	54	DATADEF	CZAEA3	CZAEAR
85	55	DDCALL	CZAFS2	CZAFSR
86	56	ABEND	CZACP1	CZACPR
87	57	CARD	CZABD7	CZABDR
88	58	TAPE	CZABD9	CZABDR
89	59	LIST	CZABD3	CZABDR
90	5A	CATALOG	CZAEI2	CZAEIR
91	5B	UNCATLG	CZAEJ5	CZAEJR
92	5C	DSCOPY	CZAFV2	CZAFVR
93	5D	TEMP DATADEF	CZAEA5	CZAEAR
94	5E	WTL	CZABQ1	CZABQR
95	5F	USATT	CZASA6	CZASAP
96	60	FINDJFCB	CZAEB1	CZAEBR
97	61	CLATT	CZASA7	CZASAP
98	62	RELEASE	CZAFJ2	CZAFJR
99	63	USAGE	CZAGB1	CZAGBP

100	64	FINDDS	CZAEC1	CZAECR
101	65	MSGWR	CZAAD3	CZAADR
102	66	UPDTUSER	CZAGC2	CZAGCR
		GENERAL SERVICES		
112	70	IOREQ	CZCSB1	CZCSBR
113	71	MSAM READ/WRITE	CZCMF1	CZCMFP
114	72	MSAM – SET UNIT RECORD	CZCMD1	CZCMDP
115	73	MSAM FINISH	CZCMH1	CZCMHP
128	80	OLTAM – DEV. ALLOC.	CZATG1	CZATGP
129	81	OLTAM – EX. I/O	CZATA1	CZATAP
130	82	OLTAM – POSTING	CZATB1	CZATBP
131	83	OLTAM – TEST COMMAND	CZATS1	CZATSP
144	90	OPEN	CZCLAO	CZCLAB
145	91	CLOSE	CZCLBC	CZCLBP
146	92	FEOV	CZCLDF	CZCLDB
147	93	RFR	CZASD3	CZASDP
148	94	GDV	CZASDX	CZASDP
149	95	AETD	CZASB5	CZASBP
150	96	OBEY	CZASA4	CZASAP
151	97	MCAST	CZATU1	CZATUP
152	98	SYSIN	CZASC7	CZASCP
153	99	LPCINIT	CZASW1	CZAMZP
154	9A	LPCEDIT	CZASW4	CZAMZP
155	9B	PRMPT	CZATJ1	CZATJP
156	9C	ATTN	CZASB2	CZASBP
157	9D	GATE	CZATC2	CZATCP
158	9E	ENTRFR	CZASD5	CZASDP
159	9F	DELENT	CZASD6	CZASDP
160	A0	CSTORE	CZCKZ1	CZCKZP
161	A1	NXTRFR	CZASD4	CZASDP
162	A2	DICTIONARY HANDLER	CZASD2	CZASDP
191 254	(reserved for TSS/360 users)			

SVCs ISSUED BY MACRO INSTRUCTIONS

SVC Code	Macro Name	Meaning
116	EXIT	normal program end
117	RAE	restore and enable interrupts
118	CLIP	read command from SYSIN (unconditional)
119	CLIC	read command from SYSIN (conditional)
120	RSPRV	restore privilege
121	ENTER	enter privileged service routine
122	RTRN	enter command language director to end run
123	DELET	enter delete program
125	PCSVC	enter program checkout subsystem
127	DLINK	transfer to dynamic loader for external symbol resolution
193	SAMPLE	sample system statistics
194	ZEROSST	zero system statistics table
195	ATTACH	attach a task to the system
201	RDI	reset drum interlock
202	LCD	indicate line code for a terminal
203	CKALOC	check for terminal MTT status
204	WAIT	wait for terminal stimuli
206	SCRTSI	special create task status index
207	CONN	connect a multiterminal task to the system
208	DCON	disconnect a multiterminal task from the system

SVC Code	Macro Name	Meaning
209	XTRTM	extract accumulated CPU time
210	SETAE	set asynchronous entry
211	SPATH	set I/O device path
212	RSTTIM	reset system time
213	XTRXTS	extract extended task status index field
214	SETXTS	set up extended task status index field
215	XTRSYS	extract system table field
216	SETSYS	set system table field
216	ALLTI	allow task initiation
216	SETYMD	set year, month, and day
216	SETTOD	set time of day
217	SETTR	set real time interval
218	REDTIM	real elapsed real time
219	ATCS	activate terminal communication subprocessor
221	RESET	reset device-suppression flag
222	PURGE	purge I/O operations
225	PRESENT	present current schedule level
226	PULSE	pulse schedule table entry level
227	CHANGE	change schedule table entry
228	SYSER	indicate nonresident-program detected error
229	TWAIT	wait for terminal I/O interrupt
231	IOCAL	I/O call
232	RJELC	remote job entry line control
233	RMDEV	remove device from task
234	ADDEV	add device to task
235	SETUP	set up task status index field
236	ADSPG	add shared virtual storage pages
237	DSSEG	disconnect shared page table from segment
238	CNSEG	connect segment to shared page table
239	XSEND	send message to another task
240	VSEND	send message to another task
241	CKCLS	check protection class
242	PGOUT	write virtual storage pages on external storage
243	TSEND	forced time-slice-end
244	SETXP	set external page table entries
245	MOVXP	move page table entries
246	XTRCT	extract task status index field
247	LSCHP	list changed virtual storage pages
248	AWAIT	wait for interrupt
249	DELPG	delete virtual storage pages
250	ADDPG	add virtual storage pages
251	SETTU	set user timer
252	DLTSI	delete task status index
253	CRTSI	create task status index
254	ERROR	indicate supervisor-detected error
254	LVPSW	load virtual program status word

CONTENTS

CONTROL BLOCKS

Data Control Block (DCB)

CHADCB

			Macro
Hex	Symbol	Meaning	Operand
0	DCBDSO	data set organization	DSORG
2	DCBMAC		MACRF
4	DCBEXL	user exit list pointer	EXLST
8	DCBDDN	ddname	DDNAME
10	DCBSYV	synad address (VCON)	SYNAD
14	DCBSYR	synad address (RCON)	
18	DCBEOV	EODAD address (VCON)	EODAD
1C	DCBEOR	EODAD address (RCON)	
20	DCBBUF	buffer length	BUFL
22	DCBEV	device type	DEVD
23	DCBBUN	number of buffers	BUFNO
24	DCBBCN	buffer control	BUFCB
28	DCBBFT	buffer technique	BUFTEK
29	DCBNCP	number of channel programs	NCP
2A	DCBREC	record format	RECFM
2B	DCBOPT	open processing option	OPTCD
2C	DCBLRE	record length	LRECL
30	DCBBLK	blocksize	BLKSIZE
32	DCBBD1	device dependent parameters 1	
32	DCBKEY	keylength	KEYLEN
32	DCBPRT	printer space	PRTSP
32	DCBSTA	stacker select	STACK
32	DCBCOD		
33	DCBDD2	device dependent parameters 2	
33	DCBMOD	mode – reader/punch	MODE
33	DCBTRT	tape recording technique	TRTCH
34	DCBERO	error options	EROPT
35	DCBPAD	padding	PAD
36	DCBRKP	relative key position	RKP
38	DCBLPA	VAM, retrieval address	
38	DCBLPDA		
3C	DCBLPN	logical record count in block	
3E	DCBOPI	options	
3F	DCBOFG	open flags	OPFLG
40	DCBMSK	DCB mask flag	
44	DCBID	DCB identifier (*%*%)	
48	DCBCON	SAM, pointer to next JFCB in concatenated data set TAM, pointer to work area	
4C	DCBDEB	SAM or TAM, pointer to DEB VAM, pointer to RESTABL	
50	DCBLEN	DCB length	
51	DCBIFL	IO flags	IFLGS
52	DCBMCD	macro code	
54	DCBIMK		IMSK
58	DCBGTV	GET (VCON)	
5C	DCBGTR	GET (RCON)	
60	DCBPTV	PUT (VCON)	
64	DCBPTR	PUT (RCON)	
68	DCBPXV	PUTX (VCON)	
6C	DCBPXR	PUTX (RCON)	
70	DCBSLV	SETL (VCON)	
7C	DCBSLR	SETL (RCON)	
78	Access method dependent portion begins		
		78 – C7 – BSAM and QSAM 120–199	
		78 – B7 – VAM 120–183	
		78 – A7 – IOREQ 120–167	
		78 – C7 – MSAM 120–199	
		78 – C4 – TAM 120–195	

Data Event Control Block (DECB)

CHADEC

Hex	Symbol	Meaning	
0	DECECB	event control block	
	DECECM	completion flag	
1	DECBSF	BSAM flags	
2	DECSVC	AWAIT supervisor call	
4	DECTYP	operation type code	
6	DECLEN	data area length	
8	DECDCB	address of DCB	
C	DECDAD	data area address	
10	DECSAD	address of status indicators	
14	DECKAD	address of VIS key	
14	DECTAD	address of TAM terminal entry list	
14	DECVCA	VCCW list address	
19	DECSTA	status	
1A	DECSB0	sense byte 0	
1B	DECSB1	sense byte 1	
1C	DECRES	TAM – response	
1C	DECVCL	VCCW list–double word length	
1D	DECCSC	TAM character set code	
1D	DECVCS	number of double words to start CCW from VCCW origin	
1E	DECFL1	flags	Bits
			7
			6
			5 active
			4 user error
			3 input area overflow
			2 buffer overflow
			1 system error
			0 intervention request
1F	DECFL2	flag	Bits
			7 abend request
			6
			5 request synad
			4 in use
			3 write
			2 read
			1 response
			0 attention
20	DECCSW	channel status word	
28	DECASB	sense bytes 0–7	

General Queue Entry (GQE)

CHAGQE

Hex	Symbol	Meaning
00	GQEFWD	forward link
04	GQETSI	TSI pointer
08	GQESVC	SVC-IORCB-MCB pointer
0C	GQESAT	set address table pointer
10	GQERR	I/O error count
11	GQEF0	flag
12	GQEF5	flag
13	GQEF4	flag
14	GQEPCB	PCB pointer
18	GQECNT	PCB count
19	GQEF1	flag
1A	GQEF2	flag
1B	GQEF3	flag
1C	GQEQPS	queue processor strings
24	GQESPT	SPT pointer

28	GQESNS	sense data
30	GQECSW	CSW
38	GQEDEV	symbolic device
3A	GQEINT	interrupt code
3C	GQEREV	reverse link

GQE Flags

11	GQEFO	80 – control unit end
		40 – sense data present
		20 – 2nd TSEND pg scan requir
		10 – paging interrupt
		08 – I/O purged
		04 – ignore device end
		02 – skip I/O request
		01 – PCB reposting
19	GQEF1	80 – paging in
		40 – paging out
		20 – VAM or system paging
		10 – waiting on sense
		08 – seek argument table setup
		04 – VAM read-after-write check flag
		02 – 0=DIRECT SVC; 1=EXECUTE SVC
		01 – IORCB associated with GQE
1A	GQEF2	80 – IORCB sense op pending
		40 – path
		20 – core block
		10 – I/O paging in
		08 – I/O
		06 – CPU number
		01 – forced TSE
1B	GQEF3	80 – shared page table
		40 – queue error
		20 – master flag
		10 – TWAIT
		08 – ASAOP process flag
		04 – path error
		02 – halt I/O
		01 – awaiting device end
13	GQEF4	80 – DRAM IORCB posting req
		40 – CEASS waiting on busy pointer
		20 – shared page migration
		10 – shared page posting
		08 – pointer to GQELOG or GQEIGQ is present
		04 – reactivate interruption
		02 – partially processed
		01 – TWAIT pageout in progress
12	GQEF5	80 – data recording
		40 – data recording restart
		20 – end of tape request
		08 – end of file
		04 – rewind/unload

Interrupt Log

RESSUP

CEAJIL

00	CPUID
	CPU1=80
	CPU2=40
	CPU3=20
	CPU4=10
01	interruption type
	18 – external
	20 – SVC
	28 – program
	30 – machine check
	38 – I/O

02	interrupt code or symbolic device address
04	interruption pointer a) for TSS I/O interruption is supervisor state, byte 4=byte in extended PSW, bytes 5-7= instruction address b) for RSS program and I/O interruptions, pointer to TSSLOG where interrupt is logged c) for all other interruptions, address of TSI at time of interruption
08	old PSW or CSW

Interrupt Log

TM

CZCJTL

03	01 – program 02 – SVC 03 – external 04 – asynchronous I/O 05 – timer 06 – synchronous I/O
04	old VPSW

Input/Output Request Control Block (IORCB)

CHAIOR

0	IORSV	SVC for IOCAL
2	IORCSB	CSW channel status byte
3	IORF3	IORCB flag byte 80 – IORCE flag 40 – IOROB flag
8	IORLN	length of IORCB in 64 byte units
9	IORGL	length of page list in doublewords
A	IORPO	relative origin of page list
B	IORKY	protection key
C	IORSF	SIO failure count
D	IORCL	length of CCW in doublewords
E	IORCS	relative origin of CCW list in doublewords
F	IORST	relative origin of starting CCW
10	IORBL	length of IORCB data buffer
11	IORBS	relative origin of data buffer in doublewords
12	IORAP	actual I/O address (2 bytes)
16	IORSD	system symbolic device address
18	IORDE	pointer to DEB
1C	IORDC	pointer to DECB
20	IORPV	pointer to posting routine (VCON)
24	IORPR	pointer to posting routine (RCON)
28	IORDT	device type codes
	or	
28	IORVB	pointer to IORCB virtual storage buffer
2C	IORBA	data buffer address
30	IORSNS	sense bytes 0–7
38	IORSN	condition codes from sense
39	IORSU	sense status field
3B	IORSL	sense failure flags
3C	IORHF	HIO retry count
43	IORHE	alternate path retry count
44	IORFL	flag bytes
	IORF1	X'80' specific path mask X'40' ignore sick indicator

		X‘20’ reissue SIO mask
		X‘10’ software command chain mask
		X‘08’ error retry
		X‘04’ issue HIO mask
		X‘02’ on unit check read R0
		X‘01’ alternate track flag
45	IORF2	X‘80’ PCI equal channel/device end mask
		X‘40’ no path exists
		X‘20’ CCW specification error
		X‘10’ SIO failed mask
		X‘08’ HIO failed mask
		X‘04’ read R0 failed
		X‘02’ sense failed mask
		X‘01’ CCWs are relocated mask
46	IORF4	X‘40’ save retry count
		X‘20’ reset device mask
		X‘10’ reset suppress flag F1
		X‘08’ interrupt code stored mask
		X‘04’ IORCB chaining mask
		X‘02’ queue channel interrupt mask
		X‘01’ drum request access method mask
47	IORF5	X‘80’ incorrect length error mask
		X‘40’ had external machine check error mask
		X‘20’ multiple I/O return mask
		X‘10’ force CE/DE/PCI on first SCC
		X‘08’ force DE on first SCC
48	IORSA	sense operation code
49	IORSE	sense address
4C	IORSG	sense flags
4E	IORSH	sense count

Interrupt Storage Area (ISA)

CHAISA

6B8	ISASSA	short save area (length, 10F)		
6E0	ISALS1	long save area (nonpriv)(length,30F)		
		6E0 length of save area		
		6E4 GPR 13 save area		
		6E8 save area of called pgm		
		6EC GPR 14–12		
730	ISA1OP	old PSW		
758		privileged long save area (length, 30F)		
7D0	ISAOP	old task PSWs (length, 6D)		
800	ISANP	new task VPSWs (length, 6D)		
			Old	New
		program	7D0	800
		SVC	7D8	808
		EXT.	7E0	810
		asynchr. I/O	7E8	818
		timer	7F0	820
		synchr. I/O	7F8	828
850	ISATDT	TDT origin		
860	ISACVP	current VPSW (length, D)		
868	ISATDY	TDY pointer		
87C	ISATMP	ptr to TM PSECT		
880	ISARCB	IORCB or MCB (length, 240D)		

Message Control Block (MCB)

CHAMCB

0	MCBLNG	message length in double words
1	MCBCOD	flag byte X‘80’ reply expected
		X‘40’ reply mask
2	MCBRCD	return code for MEB
3	MCBCD1	MCB message code
4	MCBSVC	VSEND SVC

6	MCBSPR	spare space
8	MCBSND	task ID of sending task
A	MCBRCV	task ID of receiving task
C	MCBECB	address of event control block
10	MCBTXT	message text

Page Table

CHAPGT

00	PGTCBA	core block address
01		bit 4 – availability

External Page Table

CHAXPT

00	XPTXL	ext. location of page
04	XPTF1	80 – update in place
		40 – prefer paging device drum
		20 – type program or data
		10 – changed paged bit
		08 – TWAIT complete
		04 – pages assigned
		02 – shared pages
		01 – page processed by LDR
05	XPTF2	80 – temporary external address
		20 – shared XPT entry
		10 – auxiliary storage
		0F – 4-bit protection class
06	XPTPMC	page preference counter
07	XPTPH	F0–page hold count field

Resident Shared Page Index (RSPI)

CHARSP

00	RSPPTL	shared page tbl length
01	RSPPTO	SPT origin
04	RSPSPT	shared page table
06	RSPLOCK	lock byte for shared page table
07	RSPFL1	flag byte
08	RSPGQE	GQE chain
0C	RSPN	in-use page count
0D	RSPU	unused-page count
OE	RSPLNG	number of bytes assigned this SPT

Page Control Block (PCB)

CHAPCB

00	PCBIA	internal address of page
04	PCBXA	auxiliary/external address of page
08	PCBVA	VM address of page
0C	PCBF1	flag
0D	PCBF2	flag
0E	PCBF3	flag
0F	PCBF4	flag
10	PCBER	TWAIT migration XPT ptr SA
14	PCBE2	PCB entry 2
28	PCBE3	PCB entry 3
3C	PCBCA	PCB chain address

CHAPCB flags

0C	PCBF1	E0 – VAM pageout sequence no.
		10 – bypass
		0C – VM or XTSI page
		02 – null
		01 – page I/O complete
0D	PCBF2	80 – write check complete
		40 – read/write

		20 – device preference
		10 – user core release
		08 – XTSI or PSW page
		04 – preference for auxiliary disk
		02 – VM page
		01 – XTSI page
0E	PCBF3	00 – type 1 1st XTSI page
		40 – type 2 PTP
		80 – type 3 auxiliary segment page
		C0 – type 4 segment table page
		20 – TWAIT paging operation
		10– IOCAL paging operation
		08 – relocation paging operation
		04 – pageout paging operation
		02 – dispatcher paging operation
		01 – monitor sharing page operation
0F	PCBF4	80 – TSEND paging operation
		40 – suppress posting page operation
		20 – paging request by VAM
		10 – page posting read
		08 – suppress allocation

Prefix Storage Area (PSA)

CHAPSA

0E	PSAEIC	external interruption code
10	PSASIC	SVC
12	PSAPIC	program
14	PSAMIC	machine check
16	PSAIIC	I/O
18	PSAEOP	old PSW area
40	PSACSW	CSW
48	PSACAW	CAW
58	PSAENP	new PSW area
80	PSACLO	logout area
130	PSAILO	channel logout area
148	PSAISS	interrupt stacker save area
168	PSACAS	core allocation save area
188	PSATPT	TSI pointer
18C	PSAQPT	GQE pointer
190	PSADPT	DCB pointer
198	PSATPW	PSW used by dispatcher
1B8	PSADAT	drop area
228	PSARN	recovery nucleus residing
C00	PSASER	SERR bootstrap residing

BYTE 0 **SENSE BYTES**

Bit Device Type	0	1	2	3	4	5	6	7
1052, 2150	Cmd rej	Int req	Bus out	Eq chk	╳	╳	╳	╳
2540/ 1821	Cmd rej	Int req	Bus out	Eq chk	Data chk	╳	Unusual cmd	╳
1403 1443	Cmd rej	Int req	Bus out	Eq chk	Type bar	Type bar	╳	Ch 9
2400	Cmd rej	Int req	Bus out	Eq chk	Data chk	Over-run	Write-count 0	Data Convtt Chk

(Continued)

2311/ 2841	Cmd rej	Int req	Bus out	Eq chk	Data chk	Over-run	Track-cond check	Seek chk
2301/ 2820	Cmd rej	Int req	Bus out	Eq chk	Data chk	Over-run	☒	Inval addr

BYTE 1

2400	Noise	00-Non-Xst Tu 01-Not ready 10-Rdy & no rwd 11-Rdy & rwdng		7 trk	AT load point	Wrt status	File protect	Tape ind
2311/ 2841	Data chk fld	Trk over-run	End of ctl	Invalid seq	No rec found	File prot	Missing addr mrkr	Over-flow inl
2301/ 2820	Data chk in count	Trk over-run	End of cyl.	Inval seq	No rec found	File prot	Service over-run	Over-flow inl

BYTE 2

2400	Bits 0 – 7 indicate track in error						6 & 7 indicate no error or multi-error	
2311/ 2841	Un-safe	☒	Serial-izer chk	Tag line chk	Alu chk	Unsel status	☒	☒
2301/ 2820	Un-safe	Shift reg	Skew fail	Ctr chk	Comp chk	☒	☒	☒

BYTE 3

2400	R/W vrc	Lrer	Skew	Crc	Skew req vrc	☒	Bkwd status	Com-pare
2311/ 2841	Ready	On line	Read safety	Write safety	On line	End of cyl	☒	Seek incmpl
2301/ 2820	Lrc bit 0	Lrc bit 1	Lrc bit 2	Lrc bit 3	☒	☒	☒	☒

BYTE 4

2400	Echo err	Res tape unit	Read clock err	Write clock err	Delay counter err	Seq ind C	Seq ind B	Seq ind A
2301/ 2820	Seq ind 0	Seq ind 1	Seq ind 2	Seq ind 3	Seq ind 4	Seq ind 5	Seq ind 6	Seq ind 7

BYTE 5

2311/ 2841	Command in progress when overflow incomplete occurs. or Zero

2301/
2820
	Command in progress when overflow incomplete occurs.
	Write = X'05' or
	Read = X'06' Zero

Segment Table

CHASGT (length,F)

00	SGTPTL	page table length
01	SGTPTO	page table origin
		bit 31 – availability

Aux Segment Table

CHAAST (length, D)

00	ASTDA	page table location
04	ASTN	in use page count
05	ASTU	unused-page count
06	ASTM	max. allowed pages
07	ASTF	80 – variable length segment
		40 – page table in next XTSI page
		20 – page table in core
		10 – shared segment
		01 – segment assigned

System Table (SYSTAB)

CHASYS

00	SYSPEC	pointer to start dispatchable and active list
04	SYSLOW	low core threshhold (low)
06	SYSHI	low core threshhold (high)
08	SYSFL1	flags
0C	SYSRSP	RSPI pointer
28	SYSPSW	low core PSW save area
30	SYSTOD	time of day clock
38	SYSYMD	day-month-year in serial days
40	SYSFIT	pointer to first inactive TSI
44	SYSLIT	pointer to last inactive TSI
48	SYSLT	pointer to end of active list
C0	SYSRSV	pointer to supervisor's reserve core list
C4	SYSRSC	count of pages in reserve list
1D0	SYSPF	last PDD entry flag
1D4	SYSDIP	pointer to drum interface control block
1018	SYSCSW	old channel CSW
1028	SYSSDA	sense data area
10E8	SYSPCIR	PCB/IORCB address
10EC	SYSGQER	GQE address

Task Status Index (TSI)

CHATSI

00	TSIFPT	forward pointer to next TSI
04	TSINX	number of XTSI pages
06	TSIPMF	pending and mask flags
08	TSIXXL	external location 1st XTSI page
0C	TSILOC	internal location 1st XTSI page
10	TSIUID	userid
18	TSISIN	SYSIN
1A	TSISOT	SYSOUT
1C	TSITDP	task device list pointer

(Continued)

20	TSIRPOST	pointer to page reposting GQE	
24	TSITSN	TSE GQE pointer to 2nd scan	
28	TSITIC	task interrupt count	
30	TSITIP	pointers task interrupt queue entries	
5E	TSICIO	I/O requests pending count	
5F	TSICP	paging requests pending count	
60	TSILOCK	lock byte	
61	TSIFLG	10–delay	02–in execution
		08–ready	01–page wait
		04–TSE	
62	TSIF2	80–in the wall	
		40–TWAIT	
		20–terminal I/O	
		08–inactive task	
		04–conversational	
		02–XTSI out	
		01–current ring end	
63	TSIF3	80–quantum	
		40–PRI	
		20–I/O paging	
		10–ITI/PTI reset	
		08–end of time slice	
		04–real time slice end	
		02–user time required	
		01–third level	
64	TSIF4	flags	
65	TSIBSN	unused	
68	TSIIO	I/O awaiting paging pointer	
6F	TSIQCT	quantum ctr	
70	TSIGQP	master GQE pointer pageout	
74	TSITID	task ID	
76	TSIXPR	task external priority	
78	TSIPTS	pages used last time slice	
7C	TSIRVP	reverse pointer	

Extended TSI (XTSI)

CHAXTS

00	XTSUPS	VPSW
08		unused
10	XTSCRS	control regs 0-15
50	XTSGRS	general regs 0-15
90	XTSFRS	floating point regs
B0	XTSCTI	current timer value
B4	XTSUTI	user timer value
B8	XTSLTS	last time slice value
BC	XTSATI	accumulated time
C0	XTSETI	estimated time
C4	XTSTSI	pointer to TSI
C8	XTSNPG	number of pages this time slice
CA	XTSBYA	bytes available 1st XTSI page
CC	XTSPCT	page count XTSI
CE	XTSIC	task interrupt code
D1	XTSF1	80 – XTSI has auxiliary storage
D4	XTSPTF	first PTP in chain
D8	XTSPTL	last PTP in chain

CONTENTS

APPENDIX

EXPANDED I/O INTERRUPT CONTROLS

Accomplished by using 16 control registers together with interrupt information in PSW's

Control Registers

Register	Bit Position Assignments
0	segment table register (for dynamic address translation)
1	unassigned
2	translation exception address register
3	unassigned
4	extended mask registers for I/O channel masks, used with ext. PSW-bit 6:

Bits	
0-6	CCU1 channel masks
7	CCU1 (summary)
8-14	CCU2 channel masks
15	CCU2 (summary)
16-22	CCU3 channel masks
23	CCU3 (summary)
24-30	CCU4 channel masks
31	CCU4 (summary)

5	unassigned
6	used with ext. PSW bit 13

Bits	
0,1	machine check mask extensions for channel controllers
2,3	reserved
4-7	unassigned
8	extended control mode
9	configuration control bit; specifies when partitioning can take place
10-23	unassigned
24-31	external interruption masking (used with ext. PSW bit 7)

Bit	Interruption Source
24	timer
25	interrupt key
26	malfunction alert–CPU 1 (ext. sig. 2)
27	malfunction alert–CPU 2 (ext. sig. 3)
28	reserved (ext. sig. 4)
29	reserved (ext. sig. 5)
30	external interrupt–CPU 1, 2 (ext. sig. 6)
31	reserved (ext. sig. 7)

7	unassigned
8-14	partitioning sensing registers
15	unassigned

2846 EXTERNAL MACHINE CHECK INTERRUPT CODES
(associated with external interrupts)

		CABI			UABI							
		0	1	2	0	1	2	3	4	5	6	7
FAULT↓ PSW Bits	Note	21	22	23	24	25	26	27	28	29	30	31
Multiple CPU									┌── CPU			──┐
Recognition		0	0	0	1	0	0	0	1	2	3	4
									┌── CABO			──┐
CABO Parity Check		0	0	0	0	1	0	0	P	0	1	2
UABO Parity Check		0	0	0	0	0	1	0				
CABO and UABO Parity Checks		0	0	0	0	1	1	0				
Multiple Channel						┌──		Channel				──┐
Recognition		1	1	0	0	0	1	2	3	4	5	6
Storage Interface										┌─ Stor ID		─┐
Timeout	2, 4	1	0	0	0	1	0	0	0	4	2	1
Channel Interface										┌─ Chan ID		─┐
Timeout	1	1	0	0	0	0	1	0	0	4	2	1
	1, 4		CSW	1		Prot						
SAB Parity Check	5	1	Store	1	0	Key	ID	SAB	Mark			
Invalid Address (CSW)	1, 6	1	1	1	0	0	0	0	0			
Multiple Storage Select	1, 4	1	0	0	0	0	0	1	0			
Multiple CCU Faults	3	1			1							
										┌─ Chan ID		─┐
Prefix ID Parity Check						ID P	ID 1	ID 2	1	4	2	1

Notes:

1. Binary representation of recognized channel encoded.
2. Binary representation of selected storage unit is encoded (storage A = 000, storage B = 001, etc.).
3. PSW bits 22, 23, and 25-31 ignored.
4. Storage-address-check signal returned to channel.
5. Bit 22 set 1 if indicated check detected during CSW store operation.
6. Invalid storage-address-check signal always returned to channel; CCU external machine check interruption occurs only if channel in CSW-store operation.

MACHINE CHECK INTERRUPTION (for DAT unit)

In addition to machine-check capability in table, "Time Sharing System/360 Interruption Codes," the 2067 performs machine-checks on dynamic address translation unit. Additional information about machine-check interruption is stored in first byte (bits 0-7) of translation-exception address register (control register 2). A 1-bit in translation exception address register will indicate conditions:

Bit	Condition
0	More than one associative register contains identical information, or one comparing circuit at fault.
1	One of three conditions: a) hardware error occurred; successful compare achieved with virtual address higher than addresses in segment table. b) software error; program interruption 16 occurred with address bus out bits 0-7 greater than segment table register (control register 0) bits 0-7. c) software error; program interruption 17 occurred with address bus out bits 12-19 greater than page table register bits 0-7.
2	Virtual address portion of translated address just stored in associative array does not compare with virtual address that should have been stored.
3	Reset of load-valid bits in associative array unsuccessful.
4	Parity of adder sum inconsistent with predicted parity.
5	Parity of virtual address incorrect when received by associative array.
6	Parity of data word from storage incorrect when received by dynamic address translation circuitry.
7	Parity of instruction bits 8-15 incorrect when received by dynamic address translation circuitry.

EXTENDED DIRECT CONTROL

Enables direct communication of interrupt control information between two CPUs; it uses the external interrupt signal masks in control register 6 of each CPU to determine responses to control information received when interrupts occur. The meaning of external interrupt signal masks depends on the CPU in which control register 6 resides. The write-direct assembler language instruction can be used to cause only external timing signal interrupts and external starts.

Contr. reg. 6 mask bit	External signal	Meaning in CR6, CPU1	Meaning in CR6, CPU2
24	Timer	Timer interrupt allowed	Timer interrupt allowed
25	Interrupt key	Interrupt key	Interrupt key
26	2	Not used	Malfunction alert from CPU1
27	3	Malfunction alert from CPU2	Not used
28 29	4 5	Reserved for future use	Reserved for future use
30	6	External timing signal from CPU1 or CPU2	External timing signal from CPU 1 or CPU2
31	7	Not used	Not used

The control registers can be initialized by the load multiple control instruction.

Write-direct can activate external interrupts signal 6 on the CPU to which it is directed, when control register 6, bit 30, is set to 1.

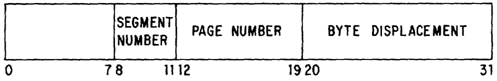
DYNAMIC ADDRESS TRANSLATION

Converts virtual storage address to physical storage addresses when the CPU is operating in extended PSW mode.

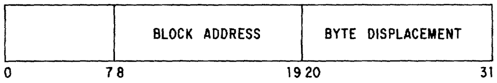
Virtual Storage Addresses (VA)

The virtual address operand is formed from the base address, index, and/or displacement, as determined by the instruction format.

VIRTUAL STORAGE ADDRESS

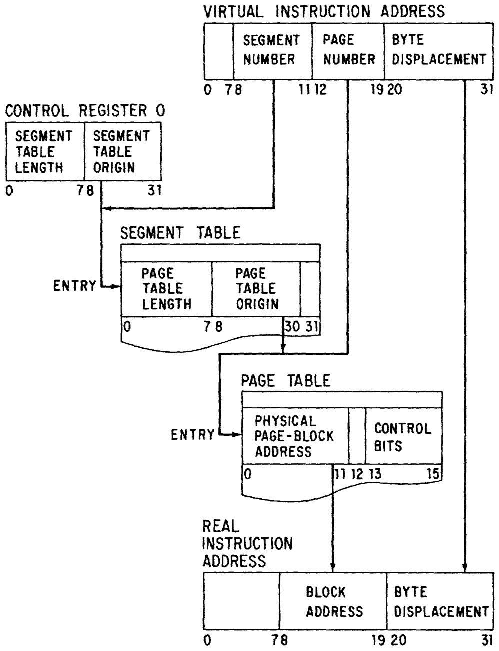


PHYSICAL STORAGE ADDRESS



Conversion Technique

1. User gets control, the origin or his segment table is placed in control register 0.
2. The virtual address is passed to the DAT unit.
   A. Segment number (from VA) + segment table origin (from control register 0) = address of segment table entry, pointing to page table origin
   B. Page table origin (from segment table entry) + page number (from VA) = address of page table entry, containing physical page block address
   C. Physical page block address (from page table entry) + byte displacement (from VA) = real instruction address



BIT ALIGNMENT OF ADDRESS ARITHMETIC

1. Computation of Segment Table Entry Address

Bits (24-bit mode)	**Meaning**	**Remarks**
8-31	Segment table origin (from control register 0)	Bits 26-31 considered 0
8-11	Added to logical address of segment table (from virtual address)	Aligned with 26-29 of segment table origin
8=31	Yields sum	Segment table entry address (30-31, always 0)

Bits (32-bit mode)	Meaning	Remarks
8-31	Segment table origin (from control register 0)	26-31 considered 0
0-11	Added to logical address of segment table (from 32-bit mode virtual address)	Aligned with 18-29 of segment table origin
8-31	Yields sum	Segment table entry address (30-31, always 0)

2. Computation of Page Table Entry Address

Bits	Meaning	Remarks
8-31	Page table origin (from segment table entry)	31 considered 0
12-19	Added to logical address of page (from virtual address)	Aligned with 23-30 of page table origin
8-31	Yields sum	Page table entry address (31, always 0)

3. Computation of Physical Address on Page

Bits	Meaning	Remarks
0-11	Real physical page address (from page table entry)	High-order portion
20-31	Displacement from virtual or logical address	Low-order portion
8-31	Physical address of instruction	Both portions

FORMATS OF REGISTERS AND ENTRIES

Bits	Meaning	Remarks
1. Segment Table Register Format		
0-7	Segment table length*	Number of 16-entry groups in segment table; all 0's = one group
8-31	Segment table origin	Segment table origin located on 64-byte boundary; bits 26-31 must be 0

2. **Segment Table Entry Format:**

0-7	Page table length	Number of entries in page table; all 0's = one entry; 1 = two entries; etc.
8-30	Page table origin	Page table origin located on 2-byte boundary
31	Page table availability	1=segment translation exception (program interrupt code 16)

3. **Page Table Entry (halfword):**

0-11	Physical block address	Starting addresses of page;
12	Page availability	1=page translation exception (program interrupt code 17)
13-15	Control bits, reserved	Must be 000 or specification exception

*Only for CPUs with 32-bit addressing feature

To avoid repetition of address translation, page table entry (physical page starting address, bits 0-11) is recorded in an associative register with, and identified by, its virtual storage address (segment and page table number, bits 8-19 of VA); eight associative registers are used by DAT unit, with this format:

ASSOCIATIVE REGISTER FORMAT

24-bit addressing

Bits	Content	Remarks
8-19	Virtual address	
20-31	Physical address	Page address from previous translation that corresponds to virtual address in bits 8-19
32-35	Unassigned	
36	Register valid	Set to 1, on loading the register
37	Recent usage, "Load"	Set to 1, on loading the register and on any use thereafter
38	Disable	Set with special diagnostic codes 8-15

32-bit addressing

same as above except for virtual address; in bits 0-19.