Systems Technology

MODEL 2086

CARD READER

INTERFACE MANUAL

ISSUED: OCT. 1970
STOCK NO.: 67094981

FAIRCHILD
SYSTEMS TECHNOLOGY

Table of Contents

Appendices

1. Mnemonics
2. Logic Equations
3. Logic Flow Chart
4. DCB and Data Formats
5. Command Set
6. Status Register
7. General Status (STST)
8. Error Status (ETST)
9. Logic Wirelist
10. I/O Cable
11. Hollerith Code to Alpha Code Transformation
12. Binary Bootstrap Card
13. B.C.D. Bootstrap for ARR
14. Card to Word Count Conversion Table

Reference

Documents

C.R.C. Major Block Diagram		95208600-20
Logic Schematics		
	C.R.C. Control A	97166107-04
	C.R.C. Control B	97166106-04
	C.R.C. Data Control	97166105-04
	C.R.C. Code Conversion	97166104-04
C.P.I. Manual		
SR-300 Manual		

1.0 Introduction

The F-24 Card Reader Control Unit (CRCU) is based on the Data Products SR 300 Card Reader. Cards may be read at a maximum rate of 300 cards/minute. Data may be read in either binary or alpha-numeric-only mode.

The CRCU is interfaced to the F-24 Central Processor through a Common Peripheral Interface (CPI) specialized as follows:

Device Address	-	040_8
Interrupt Address	-	04_8
Interrupt Priority	-	5
Memory Priority	-	5
Memory Size	-	Depends on System Configuration

The main functions of the CRCU are:

1. Data Input
   - -Accept and pack data to proper format
   - -Permit reading in alphanumeric only mode (convert Hollerith code to Alpha code and check for invalid characters)
2. Status
   - -Determine errors made by Card Reader, CPU, or programmer
   - -Hold and display status for programmer use
3. Bootstrap
   - -Provide fast means for cold starting the system

2.0 Theory of Operation

This section is a brief discussion of CRC operations. For a detailed analysis, refer to the logic flow diagram of Appendix 3.

In normal operation, the CRC is set up for a data transfer by Read SPU and a Data Control Block (DCB). Before issuing a read data SPU, the programmer will load the Accumulator with the address of the DCB. For the CRC, the DCB is a 2 word block in which the first word is the number of words to be read. The second word is the starting address in memory for data storage. Both words in the DCB should appear in octal form.

Since the Card Reader actually reads whole cards one at a time, the word count is usually the number of words that one card will hold. For binary data, this would be 40_{10} or 50_8 since two card columns of binary data fill one 24-bit computer word. For alphanumeric data, this would be 20_{10} or 24_8 since four columns (or characters) are required to fill one computer word.

Although whole cards are usually read one at a time, the programmer is not fixed to this format. The word count can vary from one to maximum core size. If the word count is less than the equivalent of one card, the number of words required will be read and stored in memory, and the rest of the card will be ignored. If the word count is greater than the equivalent of one card, cards will be continuously read until the word count is exhausted.

Initialization Phase

The CRC is prepared for data transfers during the Initialization Phase (I). Upon receipt of either Read Data SPU, the CRC is set BUSY and enters the Initialization Phase. At this time the mode of reading (Binary or Alphanumeric) is also set.

The first I phase is IO. IO is true until the next T5 time when the I counter is automatically advanced to I1.

During the I1 phase the DCB address is loaded into the CPI Memory Address Register (MAR) and a request to read from that address in memory is initiated. If no errors occur and memory access is granted (MACG), the I counter is advanced to I2 at T5 time.

During I2 the CRC Word Counter is loaded from memory, the MAR is advanced, and a new request to read from memory is initiated. If no errors accur and memory access is granted, the I counter is advanced to I 3 at T5 time.

During I3 the MAR is loaded with the starting address for data storage. If no errors occur, the I counter is advanced to I0 and the Initialization Phase is completed.

If at this time the card reader is "on" and "ready", the CRC commands the card reader to feed cards.

Binary Reading

If data is being read in the Binary mode, two columns are read and loaded into the CRC Data Buffer; a request to write the data into memory is then initiated. If access is granted in time, the data will be stored in memory and the CRC will repeat the cycle with the next two columns until the word count is exhausted.

Alphanumeric Reading

If data is being read in the Alpha-numeric mode, each 12 row column is read, converted to a 6 bit card reader code, and stored in the Data Buffer. With the data thus packed, four columns are required to fill the Data Buffer before requesting memory access. This cycle is repeated until the word count is exhausted.

There are two reasons for converting the data to a 6 bit code. One is that four characters can be stored in one computer word and thus reduce the size of core buffers. The second is that a 6 bit code yields 64 possible characters (the number of computer character codes is also 64). Note: the CRC 6 bit characters do not correspond directly to the computer 6 bit characters and must be converted through a program table look-up to TRASCII before use by other programs.

Any columns read in Alpha mode which do not map into one of the 64 legal characters will cause an error and be flagged by Invalid Character in the Status Register.

Interrupts

If the CRC has been enabled for interrupts (PON), it will issue an interrupt when a read operation is completed. If an error occurs during a read operation, the CRC will issue an interrupt immediately if the error was not a device error (i.e. - internal to the Card Reader). If the error was a device error, an interrupt will not be generated until the operator has cleared the malfunction and set the Card Reader to the "READY" condition.

Error Status

Whenever an error occurs in the CRC subsystem, the error is stored in the Status Register (refer to App. 6) and classification of the error is gated to the IB buss (refer to App. 7 and 8). The IB buss can be interrogated with STST and ETST commands. The Status Register can be interrogated by a RDS command; resetable errors will be cleared by this action.

Bootstrap Loading

The Boot function in the CRC is treated similarly to a Read One Card Binary command. If the control panel "RESET" switch is pushed, the Boot function is armed and Binary mode is selected. If the "LOAD" switch is then pushed, the Boot function is set true and the I counter starts to advance automatically with each successive T5 time. The events in the Boot Initialization Phase are different from the normal I Phase. Nothing happens during I0 or I1. During I2 the Word Counter is loaded with 50_8. During I3 the MAR is loaded with 100_8, the I phase is completed, and the Boot function is reset. The CRC then proceeds as in a normal read condition.

3.0 Logic Partioning

This section is provided as an explanation of the CRC logic in its manufactured form (i.e. - breakdown is by P.C. board). Each separable block of logic is discussed in reference to the function it performs.

3.1 Common Peripheral Interface

The CRC uses a standard Common Peripheral Interface (CPI) specialized by adapter plugs to the CRC device code, interrupt address, interrupt priority, and memory priority. The CPI, which is comprised of 3 boards, is used to interface the CRC to the memory busses and the CPU. Two of the boards are identical (except for adapters) and contain the Memory Address Register (MAR), buss to buss gating, and the option logic for specialization. The third board is used for buss control, partial command decode, and common peripheral logic functions.

The communication between the CRC and CPI is accomplished with various control lines and the Peripheral Data Buss (PDBxx). For a more comprehensive explanation of the CPI, refer to the CPI logic schematics are available, insert them here for a complete set of documentation on the CRC subsystem.

3.2 CRC Control A

The CRC Control A board is used for instruction decoding and control logic for the various states for the CRC.

Instruction Decode

The Instruction Decode network decodes all valid commands for the CRC. A list of the CRC commands is available in App. 5. Any SPU command which is invalid but does contain the CRC device code (040_8) will be flagged as such by VALC being low at SELDEV2 time. SELDEV2 is a control line from the CPI indicating that an SPU for the CRC is ready for decoding.

Bit 16 and 17 of the PDB are gated at SELDEV2 time to determine if there will be an information transfer on the accumulator bus (BNXX). If PDB17/is low there will be a buss transfer. If PDB16/ is low the transfer will be from the CRC to the CPU; if it is high the transfer will be from the CPU to the CRC.

State Control Flip-Flops

BTARM - The Boot Arm latch is used to enable the Boot flip-flop. The arming is accomplished by pushing the control panel "RESET" switch.

BOOT - The Boot flip-flop is used to initialize the CRC for a bootstrap load. It is true only during the Initialization Phase.

DBFULL - The Data Buffer Full flip-flop is used to indicate that a data word is ready for transfer to memory. It is set whenever data is strobed into the Data Buffer at Character Count 3 (CC3) time and reset when the word is transferred to memory (GDP).

MQ - The Memory Request flip-flop is used to indicate that conditions requiring access to memory are present. When access is granted MQ is reset.

DMT - The Disable Memory Transfers flip-flop is used to prohibit data transfers to memory when the CRC is not functioning properly. Either of the Read commands or Boot will reset DMT.

I - The Initialization Phase flip-flop is true whenever the CRC is being initialized for a read operation.

BSY - The Busy flip-flop is true during a read operation from start of initialization until the last word is stored in memory if no errors occur.

BINALPH - The Binary/Alphanumeric mode flip-flop is used to set the CRC in one of the two possible modes. When BINALPH is set, Binary reading is selected; when it is reset, Alphanumeric reading is selected.

FDCD - The Feed Card flip-flop is used to signal the Card Reader that it should begin feeding cards. FDCD will stay true until the Word Counter has been exhausted or an error occurs.

IOP - The Interrupt Upon Termination of Operation flip-flop is used to store the condition that interrupts are enabled and that a read operation has started. Once set, IOP will stay true until the interrupt is serviced. Note: The interrupt will not be issued (COMINT) until the present operation is completed (not BUSY or an error occurs SERR).

Miscellaneous Logic

Other logic on this board is used to communicate with the CPI. There is logic to write to memory (WMEM/), read from memory (RMEM/), and command interrupts (COMIN). There is also logic for loading the MAR from memory (LMARM/) or from the PDB (LMARD/), and logic for PDB busy (PNBSY/).

3.3 CRC Control B

The CRC Control B Board contains the Status Register logic and control logic for the CRC Data Control Board.

An explanation of the individual bits of the Status Register and the IB buss can be found in App. 6,7, and 8. If a Read Status command (RDS) is received by the CRC, the READST flip-flop will be set. The following T1 time the status will be gated to the PDB (GSTP., if the CRC is not in the Initialization phase. At the end of T1, READST is reset. During T2 time, the resetable errors in the Status Register will also be cleared by either Read data command. A general classification of the CRC status is gated to the IB buss (IB20-23) for display on the control panel indicators (>, =, <, B) in response to CRC commands.

Other logic on this board is used for gating and clocking the Word Counter, Data Buffer, Character Counter, and Initialization Phase Counter. Logic for forcing the desired word count and starting address onto the PDB for the Boot function is also located here.

3.4 CRC Data Control

The CRC DATA Control board is a generalized board to allow its use in other peripheral controllers.

Data Buffer

The Data Buffer is loaded differently for binary and alphanumeric reading. If binary reading is selected, BINALPH/ is low and I2 and I3 inputs of the multiplexers are selected. As the Character Counter advances, first the top six rows of a card column are read and then the bottom 6 rows are read. Each half-column is thus shifted into the 4 bit shift registers until 2 full columns are loaded. The data is then ready for transfer to memory (GDP). When the Character Counter equals 0 or 2, the multiplexer I3 inputs are selected and the top half-columns are loaded. When the Character Counter equals 1 or 3 the I2 inputs are selected and the bottom half-columns are loaded.

If alphanumeric reading is selected, the I0 and I1 inputs of the multiplexers are selected. Since these are tied common to the 6 Alpha code outputs, one column of encoded data is loaded with each advance of the Character Counter.

Initialization Phase Counter

The Initialization Phase Counter is a four stage ring counter. A single "ONE" is advanced one stage around the ring with each clock pulse (CPIC). When cleared, I0 is true (I0F is low).

Character Counter

The Character Counter is also a four stage ring counter. When reset, only CC0 is true. Although on different pins, CPCC and CPD are tied together on the backplane. This causes the Character Counter to advance each time a character is loaded into the Data Buffer.

Word Counter

The Word Counter is a 14 bit binary up counter. To simulate counting the Word Counter down, the inverted Word Count is loaded directly from the PDB (PDBXX/). The WC can then count up to an all "ones" state which is effectively Word Count Zero. Decode logic for WC equals zero (WCZ) and WC equals one or less (WC1L) is included.

3.5 CRC Code Conversion

The CRC Code Conversion board contains the logic for Alpha code conversion, invalid character checking, data strobe delays, line receivers, and line drivers.

Alpha Code Conversion

The code converter maps each 12 row column into a 6 bit Alpha code. This code allows only 63 legal row-punch combinations. All other punch combinations cause the code converter to output all zeros in CRCB, A, 8, 4, 2 and 1. A 64th combination is gained by allowing the character "!" (exclamation mark-row punch combination 11-2+8) to set the code converter to zeros, yet not set the Invalid Character flip-flop.

Invalid Character Checking

To insure that no Alpha type cards are mispunched, a validity checking network is included to detect invalid punch combinations. An invalid character will set the VALERR flip-flop which is a part of the Status Register. Invalid 1 is an adder-exclusive or network which detects more than one "one" in the card number field (R1-R9 excluding R8). Invalid 2 indicates a "ONE" is in both zone R11 and zone R12. Invalid 3 checks conditions on the quasi-zone R0. Invalid 4 checks all other illegal combinations. EXMRX/ allows the character "!" to be coded as zeros whithout causing a validity error.

Data Strobe Delays

A four stage one-shot string provides the necessary interface timing for data input.

Line Receivers - 18

The line receivers for the card reader signals provide a 1K Ohm termination to ground. The signal and its paired ground are on opposite sides of the board to allow twisted pair wire to be brought as close as possible on the backplane.

Line Drivers - 2

The line drivers. A ground pin on the opposite side of the board is provided for twisted pair output.

Card Reader Controller

Mnemonics

ARD	-	Alternate Read command, sets the CRC in Alphanumeric-only mode.
BINALPH	-	(ff) When set, CRCU is in Binary mode. When reset, CRCU is in Alphanumeric-only mode.
BOOT	-	(ff) Used to bootstrap load the system
BOOTSW	-	Bootstrap load switch (from control panel)
BTARM	-	(latch) used to enable the bootstrap function
BSST	-	(ff) CRCU is busy or processing an interrupt
BSY	-	(ff) CRCU is busy
CCn	-	Character Counter, character n
CDIO	-	Command an I/O operation, Read or Alternate Read
CDJAM	-	Card is jammed in card reader
CDPR	-	Card presence, a card is in the card reader station
CLEAR	-	Logic initialize, from control panel RESET switch
CLRER	-	Clock pulse to Character Counter, same as CPD
CPD	-	Clock pulse to Data Buffer
CPIC	-	Clock pulse to Initialization Phase Counter
CPWC	-	Clock pulse to Word Counter
CRCn	-	Card Reader Alpha Code, leven n(n=1,2,4,8,A,B)
CRCU	-	Card Reader Control Unit
DAVOV	-	(ff) Data Overflow, CRCU could not get memory access in time
DBFULL	-	(ff) Data Buffer is full
DCBERR	-	(ff) Data Control Block error, memory address requested is out of range
DEVERR	-	Device error, card reader has an internal error
DINDXn	-	Delayed Index pulses, n=1,2
DMT	-	(ff) Disable Memory Transfers

ETST	-	Error Test command
EXMRK	-	Exclamation mark card code has been detected
FDCD	-	(ff) Feed card level to card reader
FDCK	-	(ff) A feed check has occurred in the card reader
FDERR	-	Feed error signal from card reader
GDP	-	Gate the Data Buffer to the P-Buss
GPWC	-	Gate the P-Buss to the Word Counter
GSTLAT	-	Gate the Status Latch output
GSTP	-	Gate the Status Register to the P-Buss
I	-	(ff) CRCU Initialization Phase Mode
In	-	Initialization Phase n
IDLENOR	-	The CRCU is idle with no error
IDLERR	-	The CRCU is idle with errors
INA	-	Interrupt function is enabled (from CPI)
INDX	-	Index pulse, a column of data is ready from the card reader
INVALID	-	An invalid alphanumeric character has been read
INVCOM	-	(ff) An invalid command has been detected
IOP	-	(ff) Interrupt is required upon termination of present operation
KLRCC	-	Clear Character Counter (Same as BSY)
KLRIC	-	Clear Initialization Phase Counter
KLRSTAT	-	Clear Status Register
LMARD	-	Load Memory Address Register from the P-buss (to CPI)
LMARM	-	Load Memory Address Register from memory (to CPI)
MACG	-	Memory access granted (from CPI)
MOVFLO2	-	Memory address requested is out of range (from CIP)
MPRO	-	Memory protect switch - Inhibits write memory (from CIP)
MQ	-	(ff) Memory access is required by the CRCU
MQROR	-	Memory request "or"
PCOMP	-	Interrupt priority routine is complete
PDBnn	-	Peripheral Data Buss, bit nn
PDCLK	-	Peripheral delayed clock
PECLK	-	Peripheral early clock

PIA	-	Priority Interrupt Acknowledged
PNBSY	-	PDB is busy
PON	-	Enable priority Interrupt command
POFF	-	Disable Priority Interrupt command
PPTn	-	Peripheral Phase time n
Rn	-	Row n of data from the card reader
RD	-	Read Command (Read Binary) to CRCU
RDCK	-	(ff) A read error has occurred in the card reader
RDERR	-	Card reader read error signal
RDS	-	Read Status command to CRCU
RDY	-	Card reader is ready (from CR)
READST	-	(ff) Read status to P-Buss
RMEM	-	A read memory operation is required (to CPI)
RQM	-	Request memory access
RS	-	Reset initialization logic
SELDEV2	-	Select this device (from CPI)
SERR	-	Some error has occurred in CRC
ST	-	Set initialization logic
STST	-	Status Test command (No-Op)
VALC	-	A valid command has been detected
VALERR	-	(ff) A character validity error has occurred
WCZ	-	Word Counter equals zero
WCIL	-	Word counter equals one or less
WMEM	-	A write memory operation is required (to CPI)

Card Reader Controller

Logic Equations

Note: (#) denotes direct set or reset of flip-flops

ARD=SPUA• SPURF• SELFEVZ• CDXF• CONA •CONE

BINALPH ↑ =RD• BSYF• (PDCLK) + (#) CLEAR

BINALPH ↓ =ARD •BSYF• (PDCLK)

BOOT ↑ =BTARM •BOOTSW• BSY •(PDCLK)

BOOT ↓ =(SERR + (13 •PPT5)• (DCLK) + (#) CLEAR

BTARM ↑ =CLEAR

BTARM ↓ =BOOT

BSST ↑ =BSY•IOPF •(PDCLK)

BSST ↓ =BSYF• IOPF• (PDCLK) + (#) CLEAR

BSY ↑ = [(BSYF •IOPF• CDIO) + (BOOT• IO• PPT4)] (PDCLK)

BSY ↓ =IF• [(SERR + WCZ MQRORF CDPRF)] •(PDCLK) +(#) CLEAR

CDIO =RD + ARD

CDJAM ↑ =CDPR•RDYF •(PDCLK)

CDJAM ↓ =(#) KLRSTAT

CLEAR =RESET

CLRER =RDCK

COMINT =RDY•IOP•(BSYF + SERR)

CPC =CPD

CPD =(BINALPH•DINDX1) + [BINALPH•(DINDX1 + DINDX2)]

CPIC =I•PPT5•PECLK•(MACG + I3 + I0 + BOOT)

CPWC =(GPWC + GCP)

CRC1 =INVALIDF•(R1 + R3 + R5 + R7 + R9)

CRC2 =INVALID F•(R2 + R3 + R6 + R7 + (RO•NF)

CRC4 =INVALIDF•(R4 + R5 + R6 + R7)

CRC8 =INVALIDF•(R8 + R9 + (R0•NF)

CRCA =INVALIDF•(R12 + R0•N) + (ZF•NF)

CRCB =INVALIDF•(R11 + R12)

DATOV ↑ =DBFULL•(DINDX1)

DATOV	↓	=KLRSTAT
DBFULL	↑	=CC3 DMTF·[(DINDX1)·BINALPHF + BINALPHF + (DINDX2) BINALPH]
DBFULL	↓	=(#) (PECLK·GDP) + CLEAR
DCBERR	↑	=MOVFLO·BSY·MQ + WCZ·13) PDCLK
DCBERR	↓	=(#) KLRSTAT
DEVERR		=CDJAM + RDCK + FDCK
DMT	↑	=PDCLK (BSYF + SERR + (WCZ·IF) + (#) CLEAR
ETST		=SPUAF·SPURF·SELDEV2·CTST·CONB·COND
EXMRK		=RS·R2·R11·R0F·R1F·R3F·R4F·R5F·R6F·R7F·R9F·R12F
FDCD	↑	=(PDCLK)·RDY·SERRF·PPT5·I3·BSY·WCZF
FDCD	↓	=(PDCLK)·(WCZ + SERR) + (#) CLEAR
FDCK	↑	=(PDCLK)·FDCD·FDERR
FDCK	↓	=(#) KLRSTAT
GDP		=MACG·IF·PPT5·DBFULL
GPWC		=PPT1·I2
GSTLAT	↑	=GSTP
GSTLAT	↓	=PPT2·PECLK
GSTP		=READST·PPT1·IF
I	↑	=(PDCLK)·(BSYF·IOPF·CDIO) + (BOOT·I0·PPT4)
I	↓	=(PDCLK)·[(I3 PPT5) + SERR)] + (#) CLEAR
IDLENOR		=RDY·BSSTF·SERRF
IDLERR		=RDY·BSSTF·SERR
INVALID		=INVALID1 + INVALID 2 + INVALID3 + INVALID4
INVALID1		=CARRYS PRODUCED BY THE SUM OF R1, R2, R3, R4, R5, R6, R7, R9
INVALID2		=R11·R12
INVALID3		=R0;N·(R11 + R12)
INVALID 4		=R8·[(R11 + R12 + R1 + R9 + Z) (R1 + R2 + R9)]
INVCOM	↑	=(#) KLRSTAT
INVCOM	↓	=SELDEV2·VALCF·(PDCLK)
IOP	↑	=(PDCLK)·BSY·INA
IOP	↓	=(PDCLK)·[(BSYF·INAF) + PIA)] +(#) CLEAR
KLRCC		=BSYF

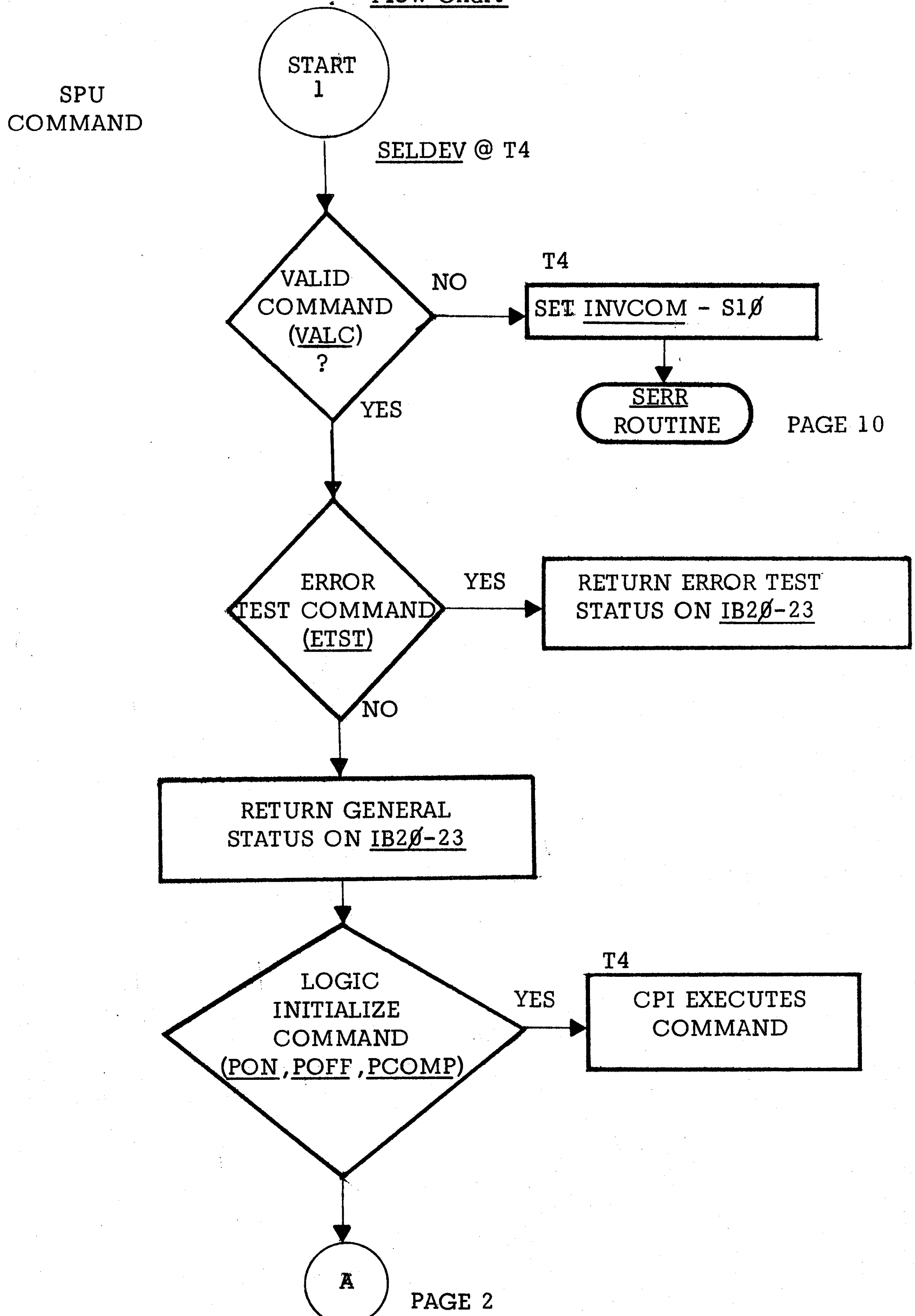
KLRIC	=CLEAR + IF
KLRSTAT	=CLEAR + CDIO·BSYF·PECLK + PPT2·GSTLAT
LMARD	=PPT1·(BOOT·I3 + BOOTF·I1)
LMARM	=PPT1·BOOTF·(I3 + MACG·WCZ·WMEM)
MQ	↑ =BSY·RQM·(PDCLK)
MQ	↓ =(PDCLK)·PPT5·MACG + (#) BSYF
MQROR	=DBFULL + MACG + MQ
PCOMP	=CLOG·CONA·COND·SELDEV2·SPUAF·SPURF
PDBOOF	=GCP·D00 + GSTP·RDY + (CPI SOURCE)
PDB01F	=GDP·DO1 + GSTP·INP + (CPI SOURCE)
PDB03F	=GDP·DO3 + GSTP·BINALPHF + BOOT·PPT1·I2 + (CPI SOURCE)
PDB04F	=GDP·D04 + GSTP·COMINT + (CPI SOURCE)
PDB05F	=GDP·DO5 + GSTP·MOVLO2 + BOOT·PPT1·I2 + (CPI SOURCE)
PDBO6F	=GCP·DO6 + GSTP·DCBERR + BOOT·PPT1·I3 + (CPI SOURCE)
PDBO7F	=GCP·DO7 + GSTP·DATOV + (CPI SOURCE)
PDBO8F	=GDP·DO8 + GSTP·RDCK + (CPI SOURCE)
PDBO9F	=GDP·DO9 + GSTP·TESTSW + (CPI SOURCE)
PDB1OF	=GDP·D1O + GSTP·INVCOM + (CPI SOURCE)
PDB11F	=GDP·D11 + GSTP·VALERR + (CPI SOURCE)
PDB12F	=GDP·D12 + GSTP·CDJAM + (CPI SOURCE)
PDB13F	=GDP·D13 + GSTP·FDCK + (CPI SOURCE)
PDB14F	=GDP·D14 + GSTP·MPROF + (CPI SOURCE)
PDB15F	=GDP·D15 + GSTP·WCZ + (CPI SOURCE)
PDB16F	=GDP·D16 + GSTP·WC1L + (CPI SOURCE)
PDB17F	=GDP·D17 + (CPI SOURCE)
PDB18F	=GDP·D18 + (CPI SOURCE)
PDB19F	=GDP·D19 + (CPI SOURCE)
PDB20F	=GDP·D20 + (CPI SOURCE)
PDB21F	=GDP·D21 + (CPI SOURCE)
PDB22F	=GDP·D22 + (CPI SOURCE)
PDB23F	=GDP·D23 + GSTP + (CPI SOURCE)
PNBSY	=I2 + I3
PON	=SELDEV2·SPUAF·SPURF·CLOG·CONB·CONE

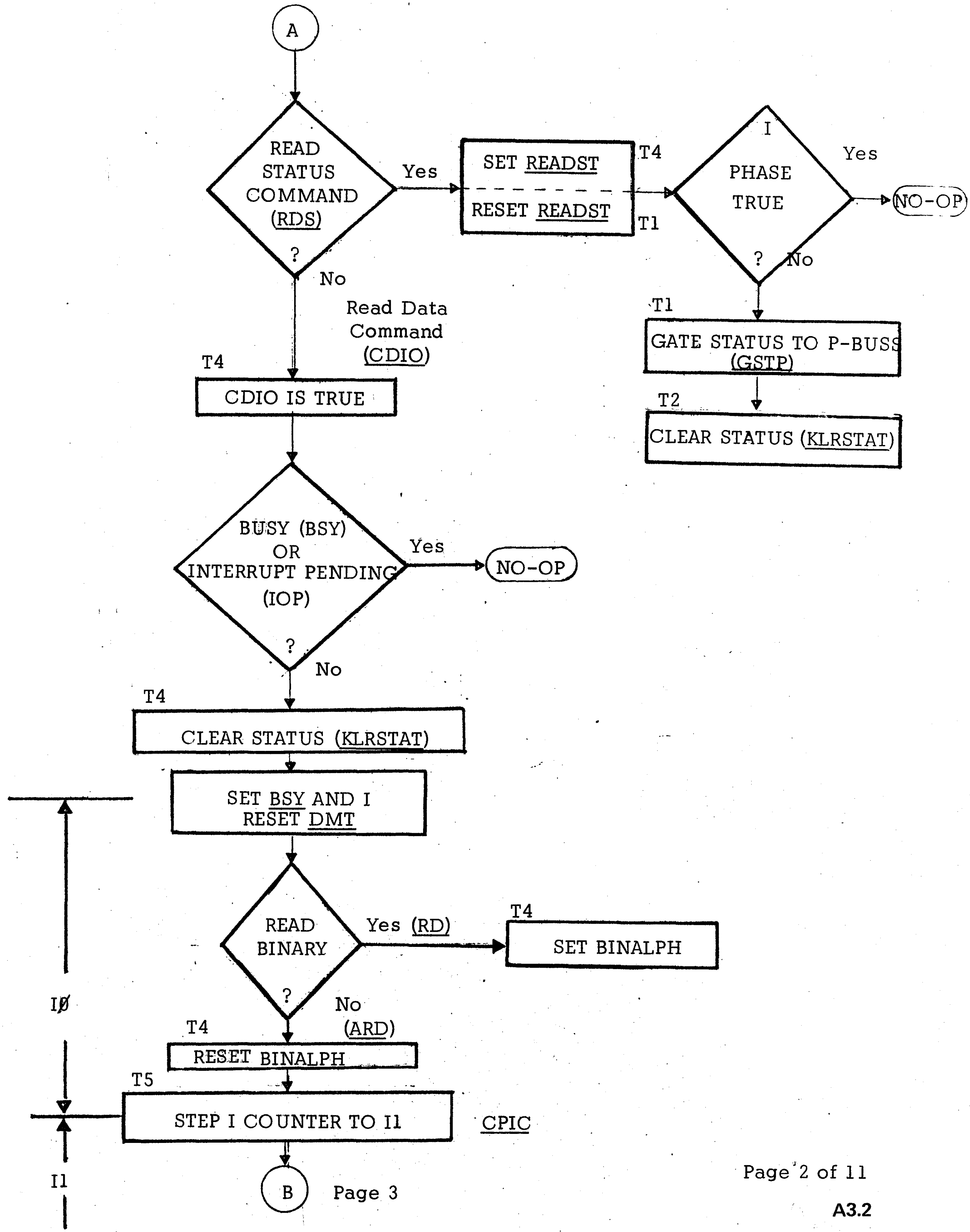
POFF		=SELDEV2· SPUAF ·SPURF· CLOG ·CONB ·CONG
RD		=SELDEV2· SPUA ·SPURF ·CDXF ·CONA ·COND
RCDK	↑	=(PDCLK)· RDERR
RDCK	↓	=(#) KLRSTAT
RDS		=SELDEV2· SPUA· SPUR ·SDXF· CONB ·COND
READST	↑	=(PDCLK) ·RDS
READST	↓	=(PDCLK) ·PPT1 + (#) CLEAR
RMEM		=I ·(MQ + MACG)
RQM		=PPT1 ·BOOTF ·(T1 + I2 + DBFULL)
RS		=SERR + (I3 ·PPT5)
SERR		=FDCK + CDJAM + VALERR + RDCK + DCBERR + DATOV + INVCOM
ST		=(BSYF· IOPF· CDIO) + (BOOT ·IO ·PPT 4)
STST		=SELDEV2 ·SPUAF ·SPURF ·CTST ·CONA ·COND
VALC		=RD + ARD + RDS + ETST + STST + PON + POFF + PCOMP
VALDAT		=GSTP
VALERR	↑	=INVALID ·EXMRKF ·BINALPHF ·(DINDX1)
VALERR	↓	=(#) KLRSTAT
WMEM		=RMEMF ·DBFULL ·(MQ + MACG)

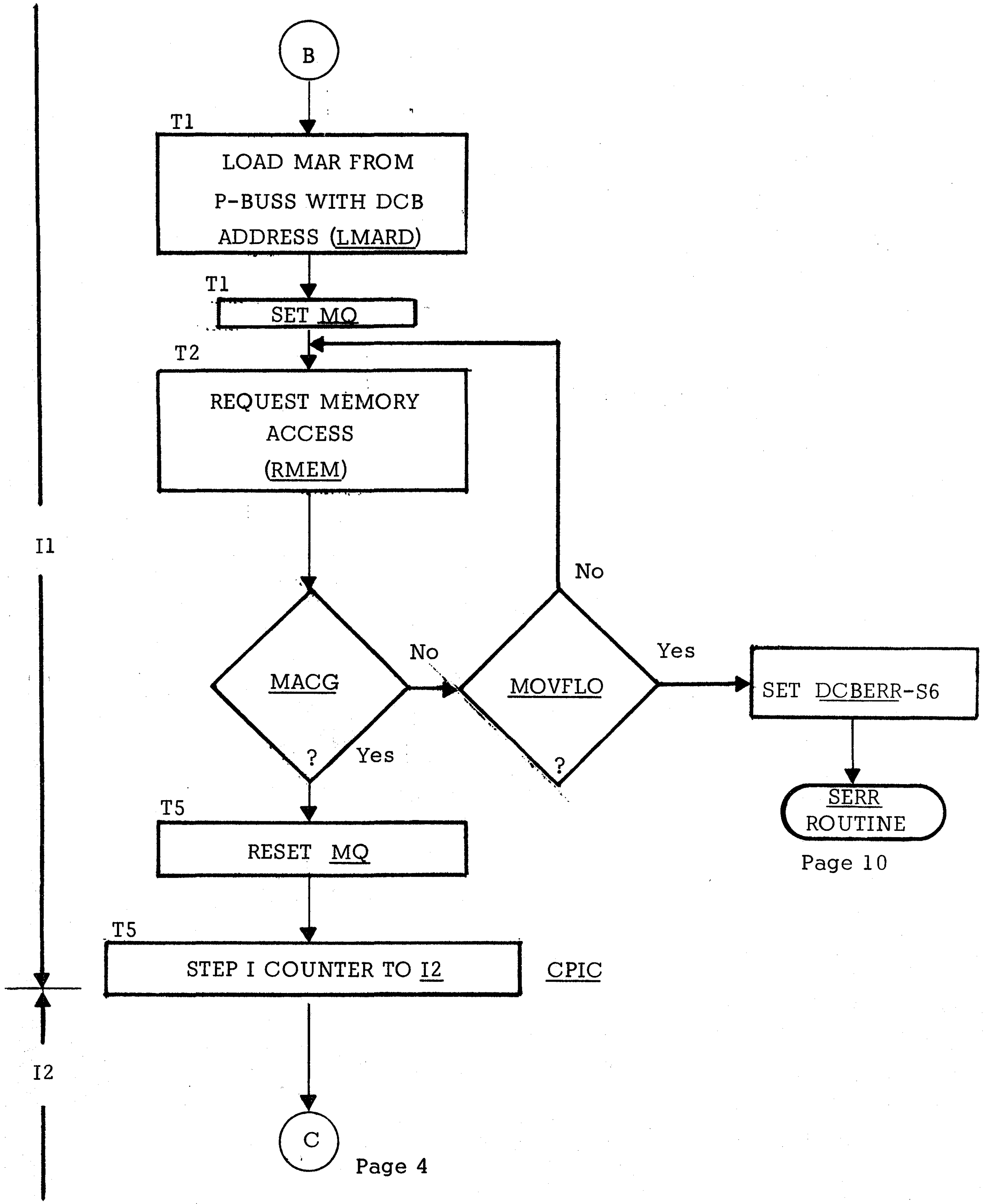
Card Reader Controller

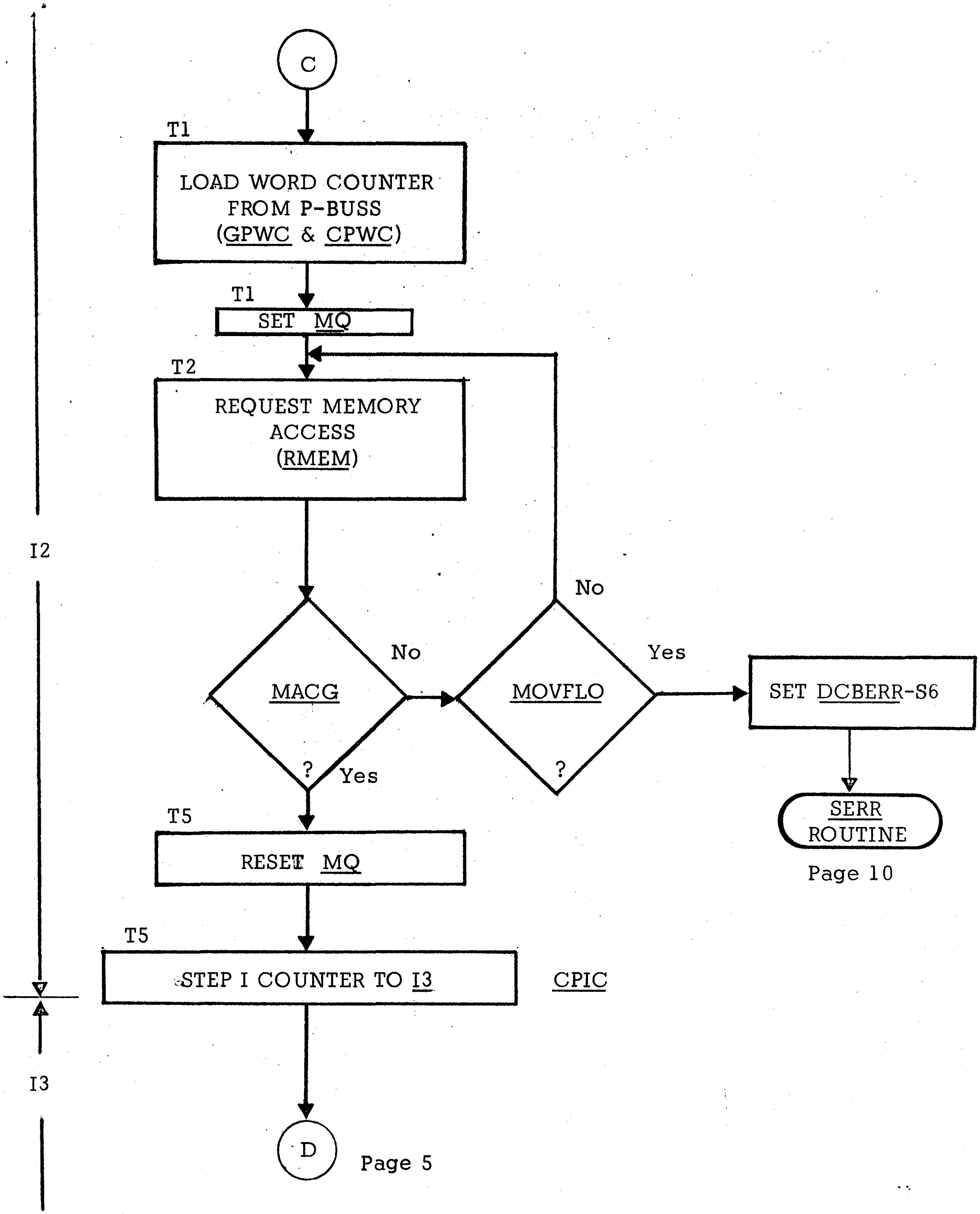
Logic

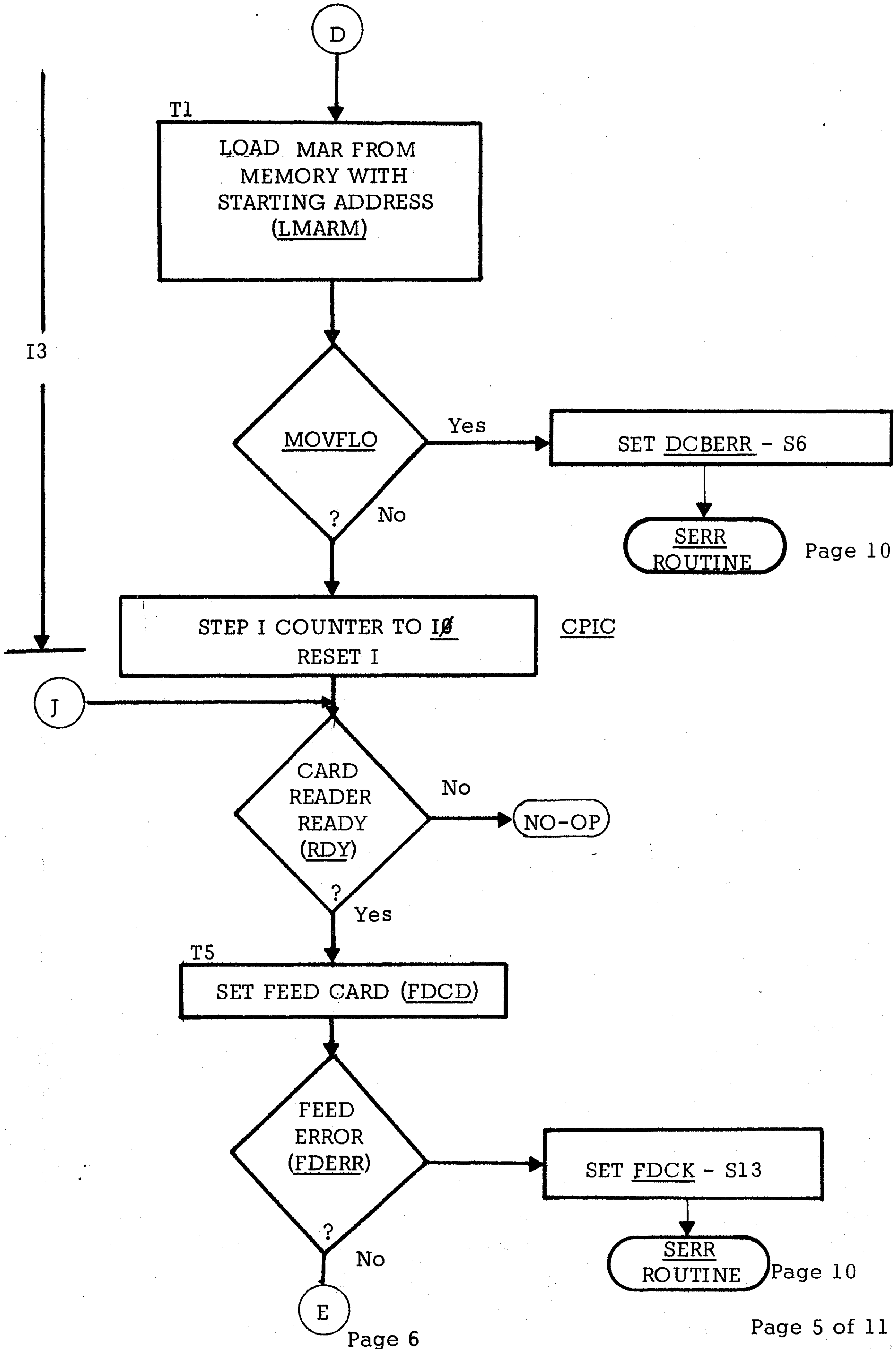
Flow Chart

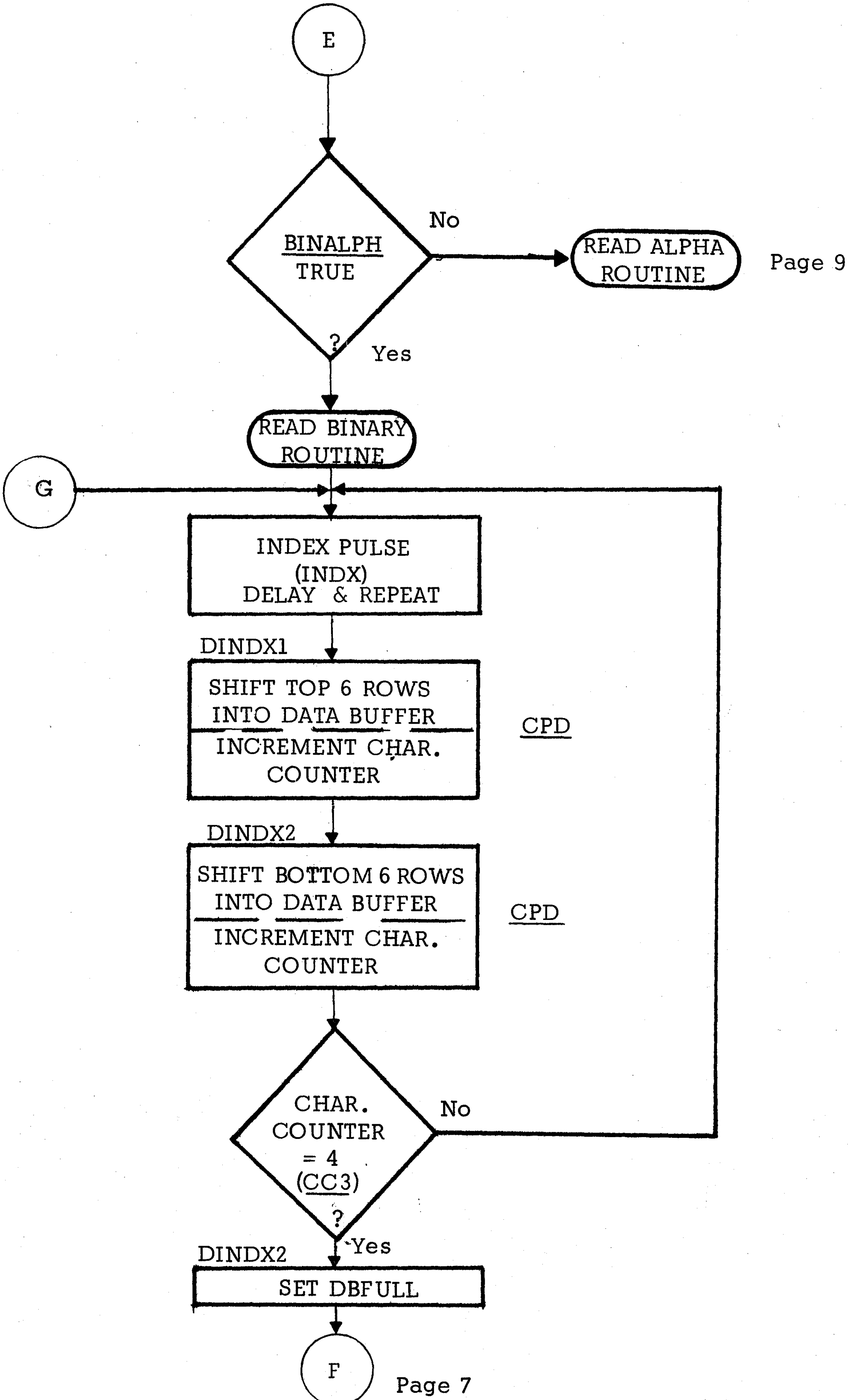


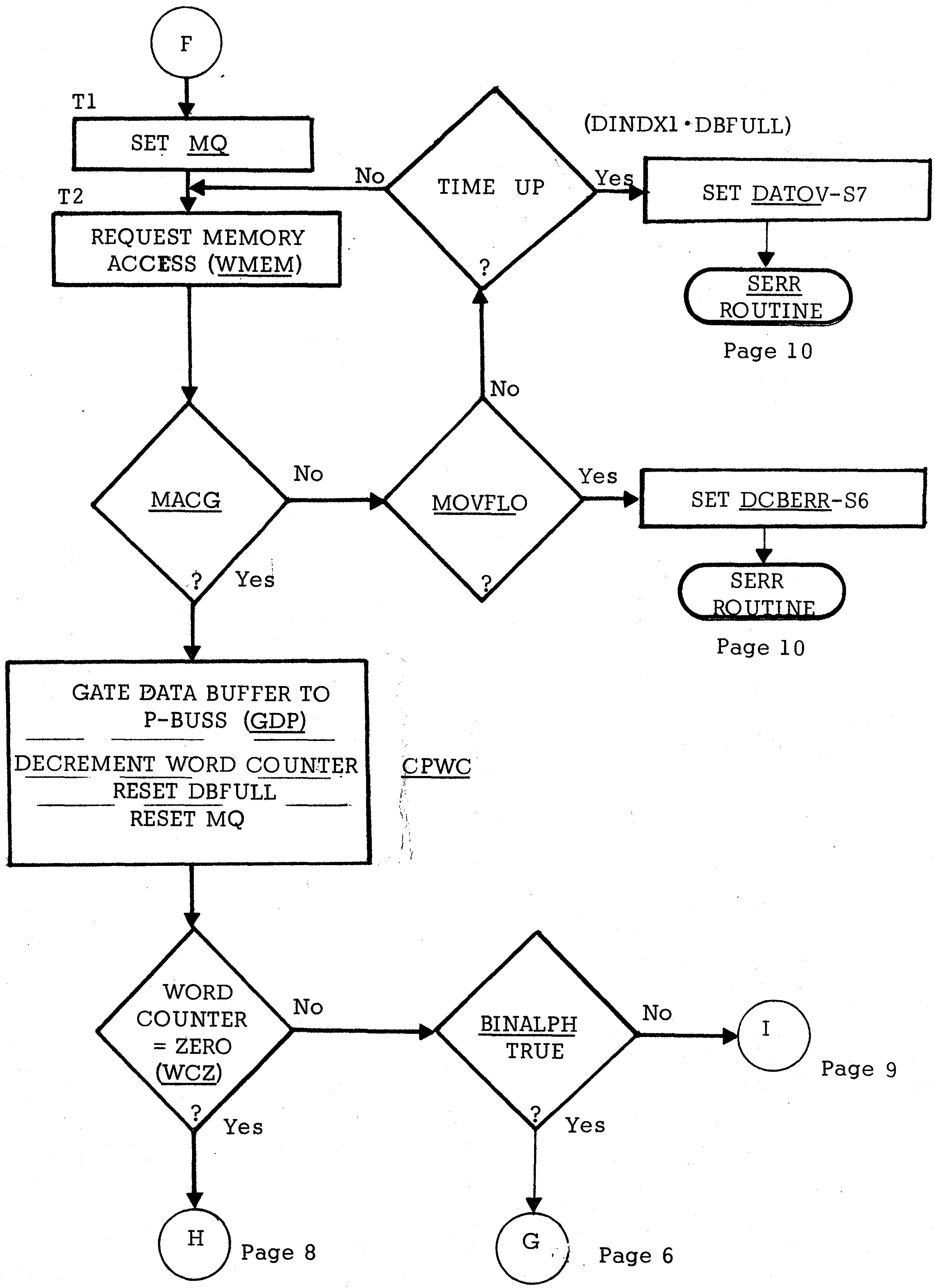

A
READ STATUS COMMAND (RDS) ?
Yes
SET READST
RESET READST
T4
T1
I
PHASE TRUE ?
Yes
NO-OP
No
T1
GATE STATUS TO P-BUSS (GSTP)
T2
CLEAR STATUS (KLRSTAT)
No
Read Data Command (CDIO)
T4
CDIO IS TRUE
BUSY (BSY) OR INTERRUPT PENDING (IOP) ?
Yes
NO-OP
No
T4
CLEAR STATUS (KLRSTAT)
SET BSY AND I
RESET DMT
READ BINARY ?
Yes (RD)
T4
SET BINALPH
No (ARD)
T4
RESET BINALPH
IØ
T5
STEP I COUNTER TO I1
CPIC
I1
B
Page 3

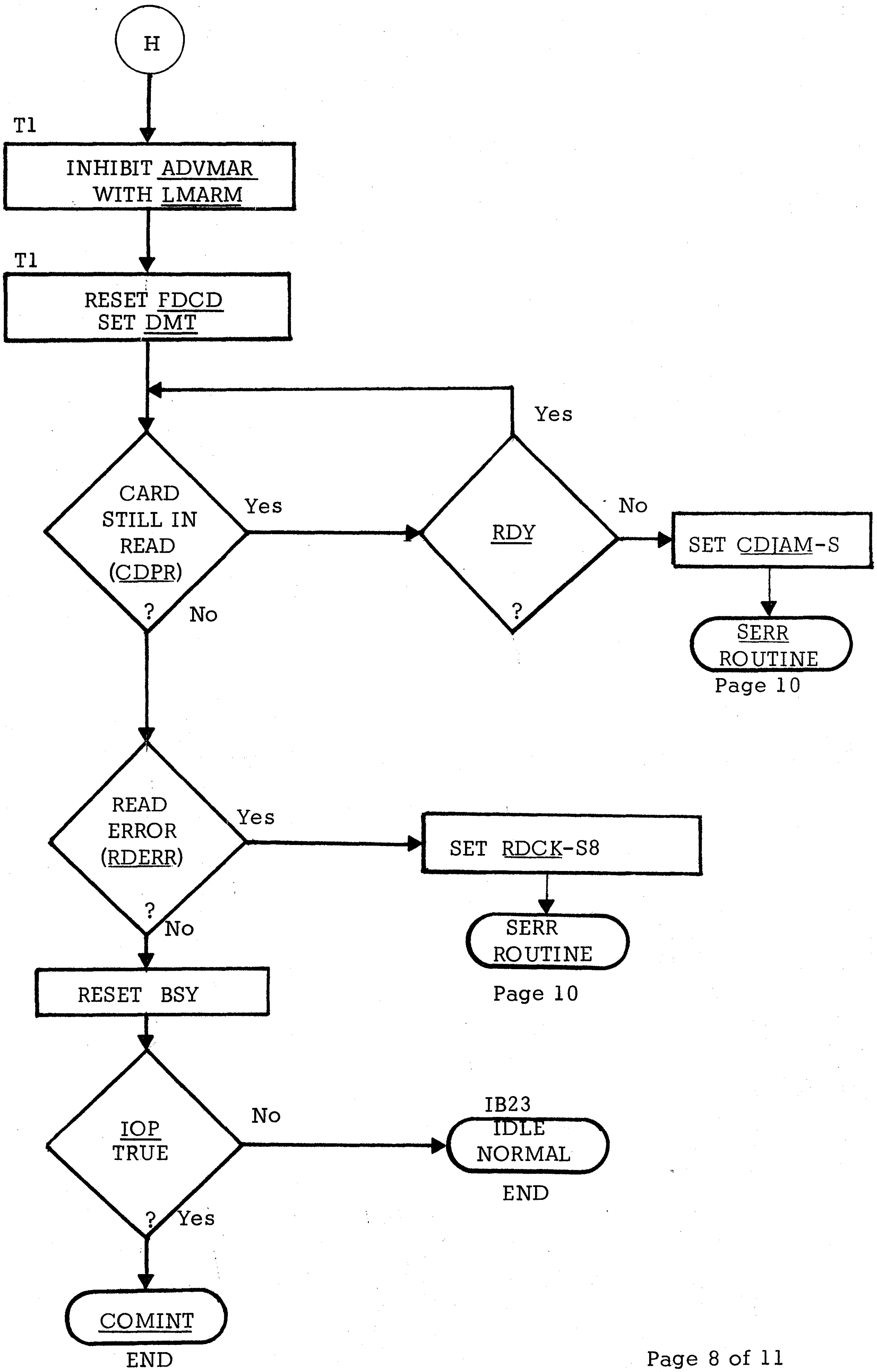

B
T1
LOAD MAR FROM P-BUSS WITH DCB ADDRESS (LMARD)
T1
SET MQ
T2
REQUEST MEMORY ACCESS (RMEM)
I1
MACG ?
No
Yes
MOVFLO ?
No
Yes
SET DCBERR-S6
SERR ROUTINE
Page 10
T5
RESET MQ
T5
STEP I COUNTER TO I2
CPIC
I2
C
Page 4


C
T1
LOAD WORD COUNTER FROM P-BUSS (GPWC & CPWC)
T1
SET MQ
T2
REQUEST MEMORY ACCESS (RMEM)
I2
MACG ?
No
Yes
MOVFLO ?
No
Yes
SET DCBERR-S6
SERR ROUTINE
Page 10
T5
RESET MQ
T5
STEP I COUNTER TO I3
CPIC
I3
D
Page 5

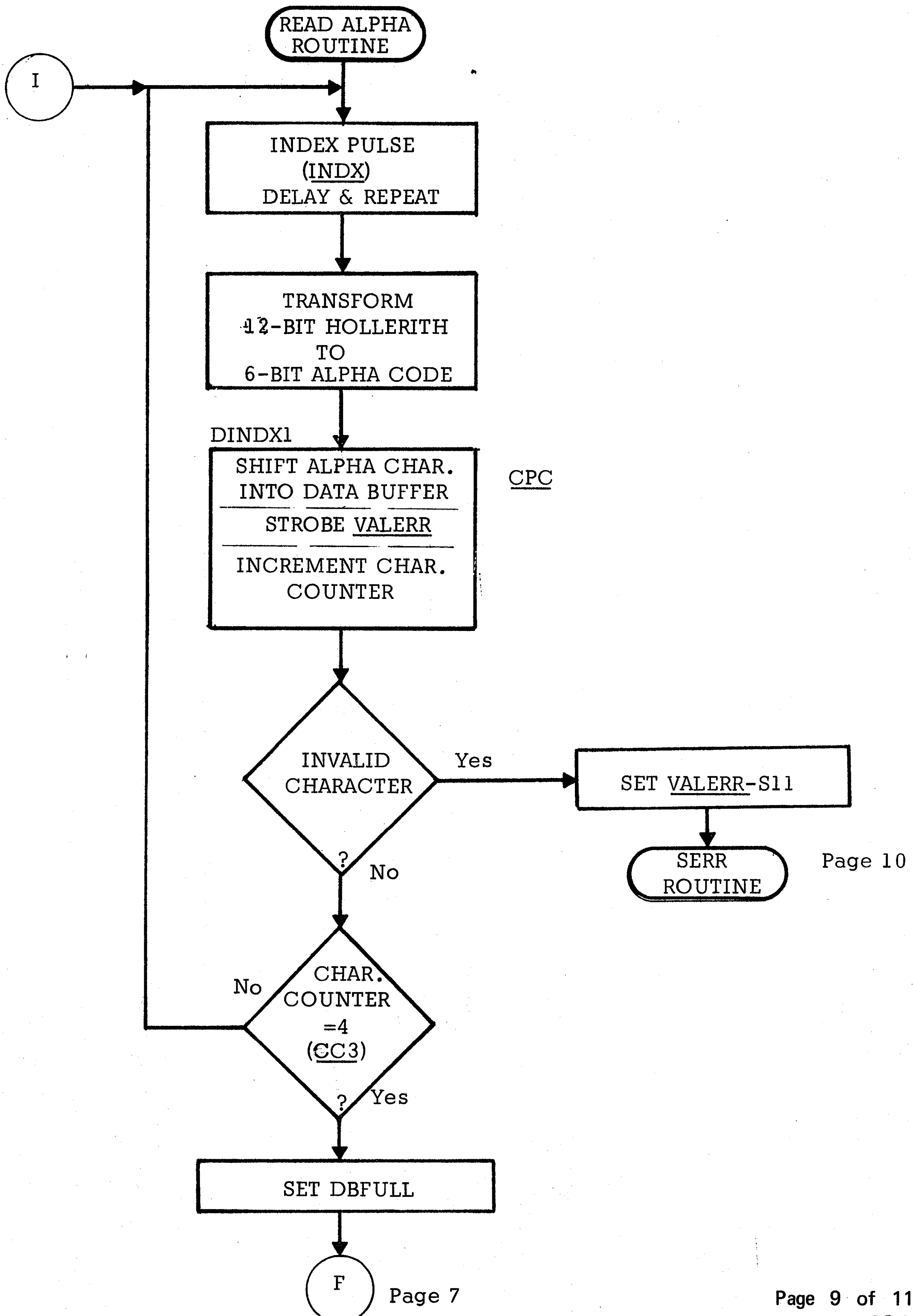

D
T1
LOAD MAR FROM MEMORY WITH STARTING ADDRESS (LMARM)
I3
MOVFLO ?
Yes
SET DCBERR - S6
SERR ROUTINE
Page 10
No
STEP I COUNTER TO IØ RESET I
CPIC
J
CARD READER READY (RDY) ?
No
NO-OP
Yes
T5
SET FEED CARD (FDCD)
FEED ERROR (FDERR) ?
SET FDCK - S13
SERR ROUTINE
Page 10
No
E
Page 6

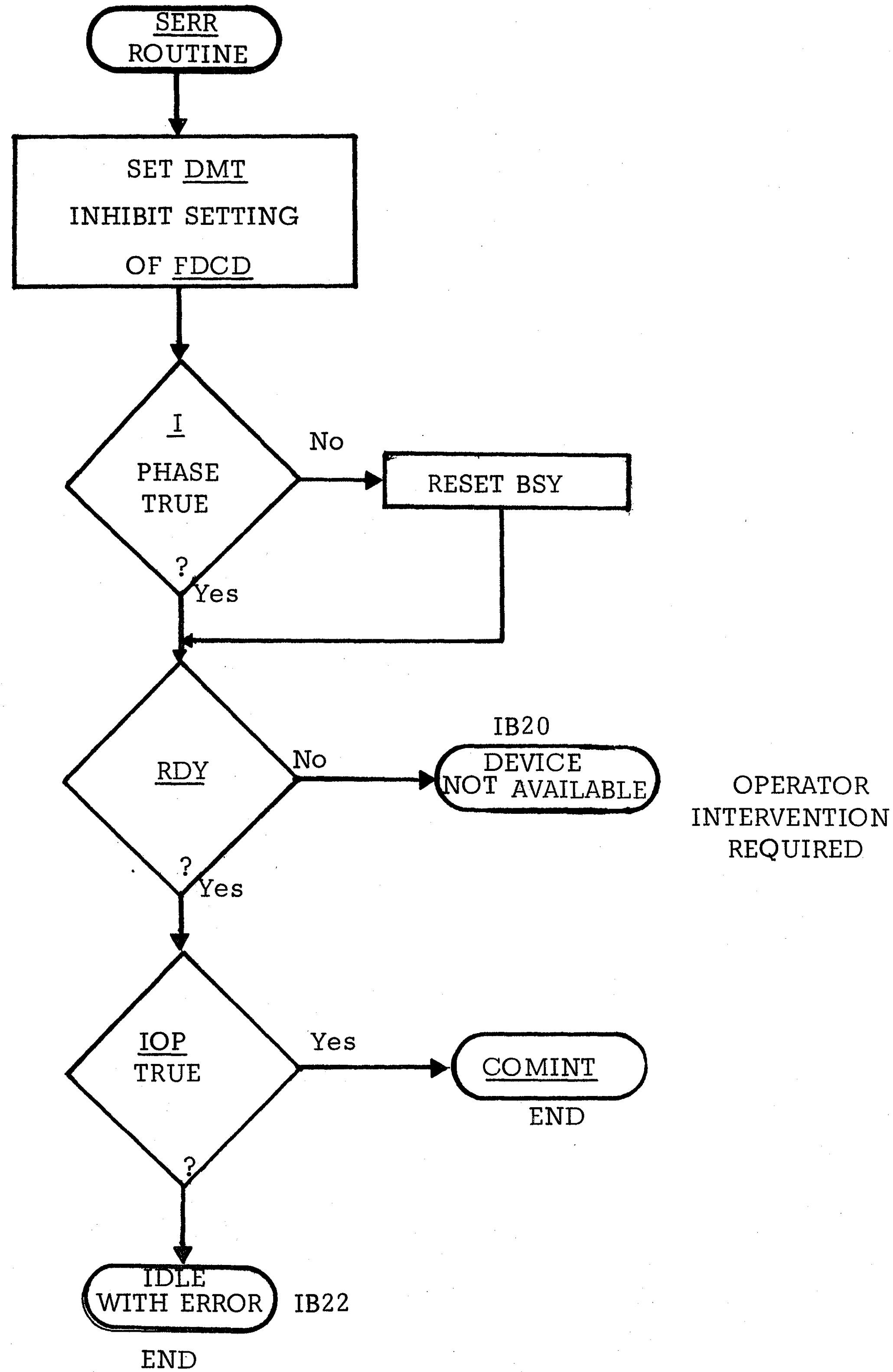

E
BINALPH
TRUE
?
No
READ ALPHA
ROUTINE
Page 9
Yes
READ BINARY
ROUTINE
G
INDEX PULSE
(INDX)
DELAY & REPEAT
DINDX1
SHIFT TOP 6 ROWS
INTO DATA BUFFER
INCREMENT CHAR.
COUNTER
CPD
DINDX2
SHIFT BOTTOM 6 ROWS
INTO DATA BUFFER
INCREMENT CHAR.
COUNTER
CPD
CHAR.
COUNTER
= 4
(CC3)
?
No
Yes
DINDX2
SET DBFULL
F
Page 7

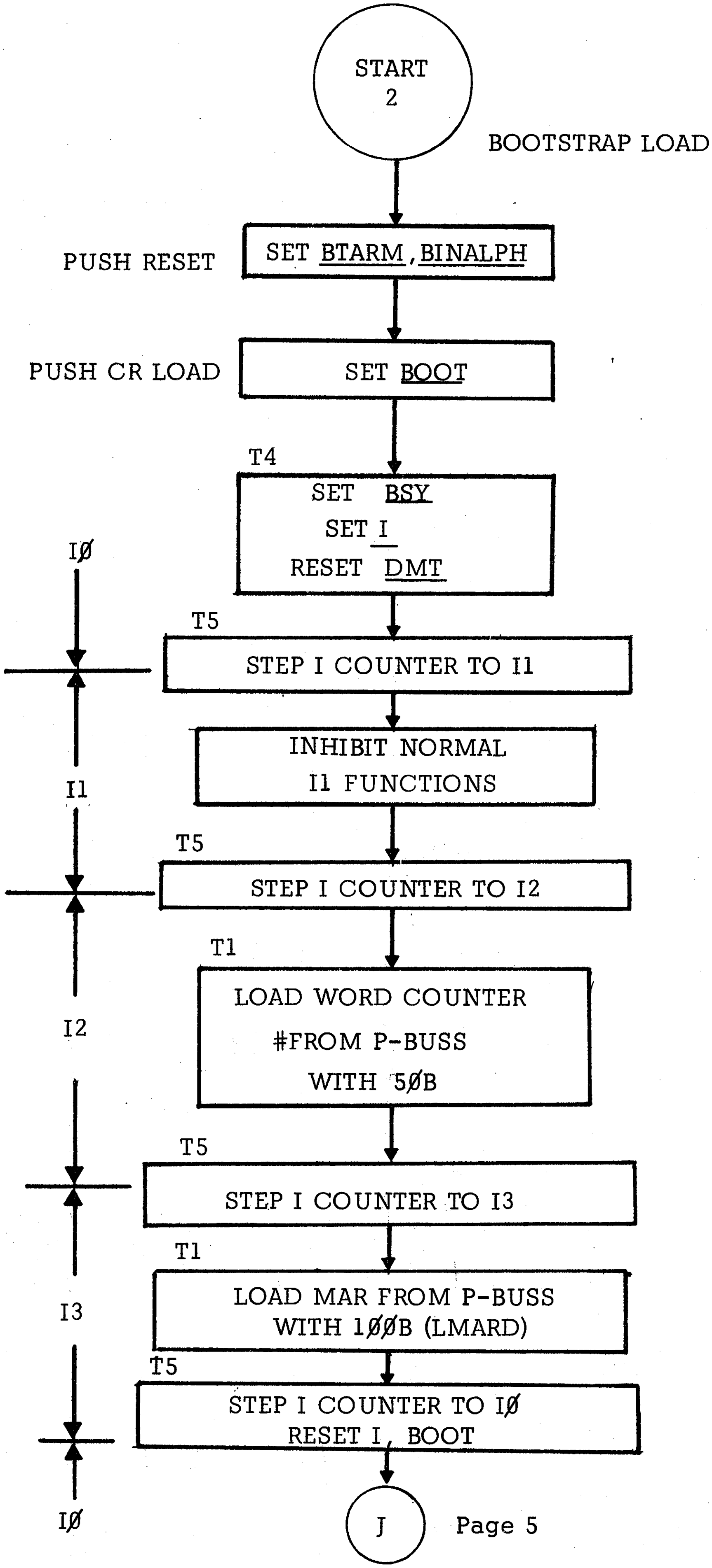

F
T1
SET MQ
T2
REQUEST MEMORY ACCESS (WMEM)
TIME UP
?
No
Yes
(DINDX1 • DBFULL)
SET DATOV-S7
SERR ROUTINE
Page 10
MACG
?
No
Yes
MOVFLO
?
No
Yes
SET DCBERR-S6
SERR ROUTINE
Page 10
GATE DATA BUFFER TO P-BUSS (GDP)
DECREMENT WORD COUNTER
RESET DBFULL
RESET MQ
CPWC
WORD COUNTER = ZERO (WCZ)
?
No
Yes
BINALPH TRUE
?
No
Yes
I
Page 9
H
Page 8
G
Page 6



H
T1
INHIBIT ADVMAR WITH LMARM
T1
RESET FDCD SET DMT
CARD STILL IN READ (CDPR) ?
Yes
No
RDY ?
Yes
No
SET CDJAM-S
SERR ROUTINE
Page 10
READ ERROR (RDERR) ?
Yes
No
SET RDCK-S8
SERR ROUTINE
Page 10
RESET BSY
IOP TRUE ?
No
IB23 IDLE NORMAL
END
Yes
COMINT
END


READ ALPHA ROUTINE
I
INDEX PULSE (INDX) DELAY & REPEAT
TRANSFORM 12-BIT HOLLERITH TO 6-BIT ALPHA CODE
DINDX1
SHIFT ALPHA CHAR. INTO DATA BUFFER
STROBE VALERR
INCREMENT CHAR. COUNTER
CPC
INVALID CHARACTER ?
Yes
No
SET VALERR-S11
SERR ROUTINE
Page 10
CHAR. COUNTER =4 (CC3) ?
No
Yes
SET DBFULL
F
Page 7


SERR
ROUTINE
SET DMT
INHIBIT SETTING
OF FDCD
I
PHASE
TRUE
?
No
RESET BSY
Yes
RDY
?
No
IB20
DEVICE
NOT AVAILABLE
OPERATOR
INTERVENTION
REQUIRED
Yes
IOP
TRUE
?
Yes
COMINT
END
IDLE
WITH ERROR
IB22
END


START
2
BOOTSTRAP LOAD
PUSH RESET
SET BTARM, BINALPH
PUSH CR LOAD
SET BOOT
T4
SET BSY
SET I
RESET DMT
IØ
T5
STEP I COUNTER TO I1
INHIBIT NORMAL
I1 FUNCTIONS
I1
T5
STEP I COUNTER TO I2
T1
LOAD WORD COUNTER
#FROM P-BUSS
WITH 5ØB
I2
T5
STEP I COUNTER TO I3
T1
LOAD MAR FROM P-BUSS
WITH 1ØØB (LMARD)
I3
T5
STEP I COUNTER TO IØ
RESET I, BOOT
IØ
J
Page 5

Card Reader Controller

DCB and DATA FORMATS

1. DCB Address Word (DCBP)

23 14	13 0
IGNORED	DCB ADDRESS

The DCB Address Word should be stored in the A register prior to issuing an SPU "READ BINARY" or "READ ALPHA" instruction. Octal format is required.

2. Word Count

The Word Count is stored at "DCB Address".

23 14	13 0
IGNORED	WORD COUNT

The Word Count can be any number from 1 to maximum residual core (octal). If the Word Count is 50_8 (40_{10}) for Binary mode, or 24_8 (20_{10}) for Alpha mode, one card of 80 columns will be read and stored in memory. If the Word Count is greater than the limits for one card, additional cards will be read and stored until the word count is satisfied. A Word Count greater than residual core will halt the read operation at the point of overflow and will be indicated by "DCB Error, S06" and "Memory Overflow S05".

3. Start Address

The Start Address is stored at "DCB Address" +1.

23 14	13 0
IGNORED	START ADDRESS

The Start Address is the location where the first word read will be stored. If the starting address is greater than maximum core size, the operation is aborted and flagged by "DCB Error, S06" and "Memory Overflow, S05".

4. Data, Binary Mode

Two card columns are read and the data is stored, bit for bit, into one word of memory as shown below. A hole punched in the card is stored as a "1" in memory.

23 12	11 0
COLUMN 1 (MOD 2)	COLUMN 2 (MOD 2)
12, 11, 0, 1, 2...ROW...8 9	12, 11, 0, 1, 2...ROW...8 9

5. Data, Alpha Mode

23 18	17 12	11 6	5 0	
COLUMN 1 (MOD 4)	COLUMN 2 (MOD 4)	COLUMN 3 (MOD 4)	COLUMN 4 (MOD 4)	
B A 8 4 2 1	B A 8 4 2 1	B A 8 4 2 1	B A 8 4 2 1	← CODE LEVEL

Four card columns are read and the data is stored into one word of memory as shown. The hole punch combination in each column is encoded to a 6 Bit code by the CRCU as defined in Appendix 11.

Card Reader Controller

Command Set

The CRCU Command Set consists of eight commands:

TYPE	COMMAND	ASSEMBLER FORMAT	OCTAL CODE
1.Ø Test			
1.1	No-op Test	STST	Ø6ØØØØ4Ø
1.2	Error Test	ETST	Ø6Ø1ØØ4Ø
2.Ø Logic Initialization			
2.1	Priority On	PON	Ø6Ø13Ø4Ø
2.2	Priority Off	POFF	Ø6Ø11Ø4Ø
2.3	Priority Complete	PCOMP	Ø6ØØ1Ø4Ø
3.Ø Data Transfer			
3.1	Read Binary	RD	Ø64Ø144Ø
3.2	Read Alpha	ARD	Ø64Ø344Ø
3.3	Read Status	RDS	Ø661144Ø

COMMAND ACTION

1.1 No-op Test

The CRCU will indicate General Status on the console Indicators. See Appendix 7 for status interpretation.

1.2 Error Test

The CRCU will indicate Error Status on the console Indicators. See Appendix 8 for error interpretation.

2.1 Priority On

The CRCU will indicate General Status and the CPI Interrupt Enable (INA) flip-flop (S02) will be set. The CRCU can now initiate priority interrupts when other necessary conditions are met.

2.2 Priority Off

The CRCU will indicate General Status and the CPI interrupt enable flip-flop will be reset.

2.3 Priority Complete

The CRCU will indicate General Status and the Interrupt in Process (INP) flip-flop (S01) will be reset to allow peripherals with lower priority to be serviced.

3.1 Read Binary

The CRCU will indicate General Status. The Binalph flip-flop will be set and S03 will be false. If the CRCU is in the Idle state, BINARY data will be read as specified by the DCB.

3.2 Read Alpha

The CRCU will indicate General Status. The Binalph flip-flop will be reset and S03 will be true. If the CRCU is in the idle state, Alpha-numeric coded data will be read as specified by the DCB.

3.3 Read Status

The CRCU will indicate General Status. If the CRCU is not in the DCB Initialization Phase, the Status Register will be transferred to the Accumulator. The status transfer will have been valid if S23 is true. Refer to Appendix 6 for Status Register representation.

Card Reader Controller

Status Register

Bit Position	Logic Name	Meaning When Set
0	RDY	Card Reader is "READY"
1	INP	Interrupt is in process
2	INA	Interrupt is enabled
3	BINALPHF	CRCU is in Alpha mode
4	COMINT	CRCU is attempting to initiate an interrupt
5	MOVFLO2	CPI Memory Address Register contains an illegal address
6	DCBERR	Illegal memory address has occurred while processing an SPU or initial word count was zero
7	DATOV	CRCU could not get access to memory in time to service reader
8	RDCK	Card Reader has detected a read error
9	SPARE	
10	INVOCM	CRCU has detected an invalid SPU
11	VALERR	CRCU has detected and invalid character in Alpha mode.
12	CDJAM	CARD READER has detected a card jam
13	FDCK	Card Reader has detected a feed error
14	MPROF	CPI Memory Protect switch is disabled
15	WCZ	Word Counter equals zero
16	WCIL	Word Counter equals one or less
23	GSTP	Valid Status Transfer

Card Reader Controller

General Status

General Status is returned by the CRCU for all SPU commands except Error Test (ETST). This information is returned on N-buss bits 20-23 to the indicators BE, <, =, and > respectively.

Status response, as interpreted by testing indicators, is as follows:

Indicator	Meaning When Set
BE	Card Reader not available-operator Intervention required
<	Subsystem Busy - Card Read operation in process
=	Subsystem Idle/Error - Card Reader is ready, CRCU not busy, but error has been detected in previous operation
>	Subsystem Idle/Normal - Card Reader is ready, CRCU not busy, and no errors exist

Card Reader Controller

Error Status

Error Status is returned by the CRCU for the SPU command Error Test (ETST).

Status response, as interpreted by testing indicator, is as follows:

Indicator	Meaning When Set
BE	DCB ERROR - Memory address out of range
<	DATA OVERFLOW - CRCU could not get access to memory
=	Device Error - Feed error, read error, or card jam
>	VALIDITY ERROR - CRCU has detected an invalid alpha character

Card Reader Controller

Hollerith Code

To

Alpha Code Conversion

HOLLERITH	ALPHA	OCTAL TRASCII	ASCII	CHARACTER	029
NO PUNCH	20	00	240	SPACE	
11-8-2	00	01	241	!	
8-7	17	02	242	"	
8-3	13	03	243	#	
11-8-3	53	04	244	$	
0-8-4	34	05	245	%	
12	60	06	246	&	
8-5	15	07	247	'	
12-8-5	75	10	250	(	
11-8-5	55	11	251	)	
11-8-4	54	12	252	*	
12-8-6	76	13	253	+	
0-8-3	33	14	254	,	
11	40	15	255	- (MINUS)	
12-8-3	73	16	256	.	
0-1	21	17	257	/	
0	12	20	260	0	
1	01	21	261	1	
2	02	22	262	2	
3	03	23	263	3	
4	04	24	264	4	
5	05	25	265	5	
6	06	26	266	6	
7	07	27	267	7	
8	10	30	270	8	
9	11	31	271	9	
0-8-2	32	32	272	:	0-8-2
11-8-6	56	33	273	;	
12-0	72	34	274	<	12-0
8-6	16	35	275	=	
11-0	52	36	276	>	11-0
0-8-7	37	37	277	?	
8-4	14	40	300	@	
12-1	61	41	301	A	
12-2	62	42	302	B	
12-3	63	43	303	C	

HOLLERITH	ALPHA	TRASCII	ASCII	CHARACTER	029
12-4	64	44	304	D	
12-5	65	45	305	E	
12-6	66	46	306	F	
12-7	67	47	307	G	
12-8	70	50	310	H	
12-9	41	51	311	I	
11-1	41	52	312	J	
11-2	42	53	313	K	
11-3	43	54	314	L	
11-4	44	55	315	M	
11-5	45	56	316	N	
11-6	46	57	317	O	
11-7	47	60	320	P	
11-8	50	61	321	Q	
11-9	51	62	322	R	
0-2	22	63	323	S	
0-3	23	64	324	T	
0-4	24	65	325	U	
0-5	25	66	326	V	
0-6	26	67	327	W	
0-7	27	70	330	X	
0-8	30	71	331	Y	
0-9	31	72	332	Z	
12-8-4	74	73	333	[	<
11-8-7	57	74	334	\	¬
0-8-5	36	75	335	]	>
12-8-7	77	76	336	↑	|
0-8-5	35	77	337	←	—

All other hole punch combinations are illegal in the Read Alpha mode and will be flagged by "Validity Error, S11"

Card Reader Controller

Binary Bootstrap

Card

The Binary Bootstrap Card in conjunction with the Card Reader Boot function facilitates automatic loading of systems programs. The code on this card is force-loaded into memory location 100_8. The loader can then be initialized through 100_8; 24_8 words (one alpha-mode card) will be read into 50_8 and the program will be entered at 55_8.

1ØØ	24ØØØ114	ST	LDA	DCPP
1	Ø64Ø344Ø	RD	ARD	CDRD
2	Ø314Ø1Ø1		BOI	BSY,NA RD
3	Ø6ØØØ4Ø	RDZ	STST	CDRD
4	Ø314Ø1Ø3		BOI	BSY,NA RDZ
5	Ø6Ø1ØØ4Ø		ETST	CDRD
6	Ø314Ø111		BOI	<,B HLT
7	Ø32ØØ1ØØ		BOI	= ST
11Ø	Ø1ØØØØ55		BRU	55B
1	ØØØØ1ØØ	HLT	BAH	
2	ØØØØØØ24	DCB"A"	DATA	BLOCK LENGTH
3	ØØØØØØ5Ø	DCB"B"	DATA	MEM LOC
114	ØØØØØ112	DCBP	DATA	DCBA

Card Reader Controller

BCD Bootstrap for

ARR

	RELOCATED LOC	LOC	CODE		INST	COMMENTS
		5Ø	ØØØØØØ51	DCBP	DAYA	DCB"A"
		1	ØØØØØ14Ø	DCB"A"	DATA	BLOCK LENGTH
		22	ØØØØØØ6Ø	DCB"B"	DATA	MEM LOC
		3	ØØØØØ4Ø6	DCB"C"	DATA	TASA
		4	ØØØØØØØØ			
		55	Ø57ØØØØØ	ST	LDX	7, Ø
		6	Ø56ØØØØ7		LDX	6, 7
		57	247ØØØ64		LDA	7, 64
		6Ø	147ØØØ4Ø		STA	7, 40
		1	117ØØØØ1		ATX	7, 1
		2	Ø33ØØØ57		BOI	>, 57
		3	Ø1ØØØØ4Ø		BRU	4Ø
	4Ø	4	Ø661147Ø	STZ	RDS	DISC
	1	5	24ØØØØ5Ø		LDA	DCBP
	2	6	Ø64Ø147Ø		RD	DISC
CALL	3	7	Ø314ØØ41		BOI	BSY, NA 41
ARR	4	7Ø	Ø6ØØØØ7Ø		STST	DISC
	5	1	Ø314ØØ44		BOI	BSY, NA 44
	6	2	Ø34ØØ1ØØ		BOI	IDLE, 1ØØ
	7	3	Ø1ØØØØ4Ø		BRU	STZ

The BCD Bootstrap Card can be used after the Binary Bootstrap to load the Automatic Restart Routine (ARR) from the Disc. The code from this card is loaded at 50_8 and entered at 55_8. If the ARR is loaded and the teletype is on, an "*" will be typed. After the ARR is loaded it can be reentered at any time through 125_8.

Card Reader Controller

Card to Word Count

Conversion Table

This table is provided as an aid for use of the multiple card read capability of the CRC.

Number of Cards	Octal Word Count	
	Binary	Alpha
1	50	24
2	120	50
3	170	74
4	240	120
5	310	144
6	360	170
7	430	214
8	500	240
9	550	264
10	620	310
11	670	334
12	740	360
13	1010	404
14	1060	430
15	1130	454
16	1200	500
17	1250	524
18	1320	550
19	1370	574
20	1440	620
30	2260	1130
40	3100	1440
50	3720	1750
100	7640	3720
200	17500	7640
300	27340	31560
400	37200	17500

COMPUTATION EXAMPLE:

NUMBER OF CARDS TO BE READ IN BINARY IS 225_{10}.

OCTAL WORD COUNT IS

	CARDS	WC	
	200	17500	
	20	1440	
	5	310	
DECIMAL SUM	225	21450	OCTAL SUM

$\underline{21450}_8$ IS WORD COUNT