Internal Memorandum

To	Distribution	Date	July 28, 1977
From	Victor Schwartz and Roy Ogus	Location	Palo Alto
Subject	D0 RS232C Controller Functional Specifications	Organization	SDD/SD/CS

XEROX

Filed on: [MAXC]<SCHWARTZ>Controller-memo.press

Introduction

This memo specifies the functions performed by the *D0 RS232C Controller*, and how PILOT software interfaces to the controller microcode in order to control these functions.

Controller Status Block

Software and microcode communicate via *Controller Status Blocks* (CSBs) located at dedicated memory addresses. There are separate CSBs for input and output (although implementation may assign them contiguous), since the support of full-duplex links requires the microcode to treat input and output as separate channels. The CSB contains various flags (see *Table Formats* below), and a pointer to a chain of *Input/Output Control Blocks (IOCBs).*

Input/Output Control Block

The IOCB is the structure by which the software passes a single command to the controller microcode. Parameters within the IOCB describe the function to be performed, the resulting status, and the data areas involved. IOCBs are chained together by PILOT software, and are processed sequentially by the microcode. Each IOCB instructs the microcode as to which process wakeups to perform and when. Each *processed* IOCB is flagged by the microcode, and later dequeued by the software.

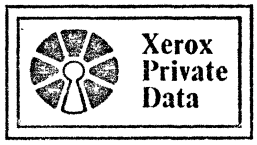
Command

Five commands are described by the command field in the IOCB. They are:

1. Reset
2. Change parameter
3. Transmit/Receive frame
4. Transmit break (asynchronous lines only)
5. Dial-a-Digit

Frames

A *frame* is the smallest meaningful unit of transmission across the communication link. The format and content of a frame depend upon the link control protocol, and are described in *Frame Formats* below, for each of the link control protocols supported by the D0 RS232C Controller. Frames are stored into frame buffers allocated by software. A single frame may be split across non-contiguous buffers to allow scatter/gather facilities, although no buffer contains more than one frame.



A single IOCB points to a single buffer. Therefore, on output, the software must indicate (in the IOCB) whether a frame starts/ends in the associated buffer. On input, the microcode will indicate (in the IOCB) whether a frame starts/ends in the associated buffer. Two additional flags in the IOCB allow the software to specify which buffers may be used by the microcode for the start of a frame, and to specify which buffers should force any overflow data for the current input frame to be discarded. These facilities allow the software to allocate buffers efficiently based on the anticipated nature of incoming data, and to *resynchronize* should the data be other than as anticipated. For example, if the software expects incoming frames to consist of a short header and a long body, which it wishes to read into separate buffers, it can allocate a small buffer exactly equal to the header length, followed by a buffer large enough to hold the largest anticipated frame body. The IOCB for the small buffer would have the flag set allowing a frame to start, and the IOCB for the large buffer would have the flag set requiring a frame to end. If a short frame arrives which fits in the small buffer (i.e. loss of synchronization between frame size and buffer size), the microcode would *bypass* the large buffer, thereby preventing it from being used inefficiently to hold a small frame. Should an illegally long frame arrive, the flag in the IOCB for the large buffer would result in truncation, preventing allocation of arbitrarily large numbers of buffers to invalid frames.

Device Dependent Parameters

In order to transmit and receive frames correctly, the controller microcode must know a number of things about the device at the other end of the communication link, as well as the associated link control protocol. This information is communicated to the microcode in two ways. First, there are three *variants* of microcode (one for asynchronous protocols, one for BSC, and one for the bit-oriented protocols such as HDLC, SDLC, and ADCCP). The software requests the loading of the proper *variants* microcode prior to any other controller activity. Each microcode *variants* is properly coded to handle the *general characteristics* of the associated line discipline. The second way to *configure* the microcode, is via parameters, listed in *Table Formats* below, which are passed to the microcode via *change parameter* commands when the link is first activated (i.e. following a *reset),* but may also be changed synchronously with frame transfers at any time.

Built-In Parameters

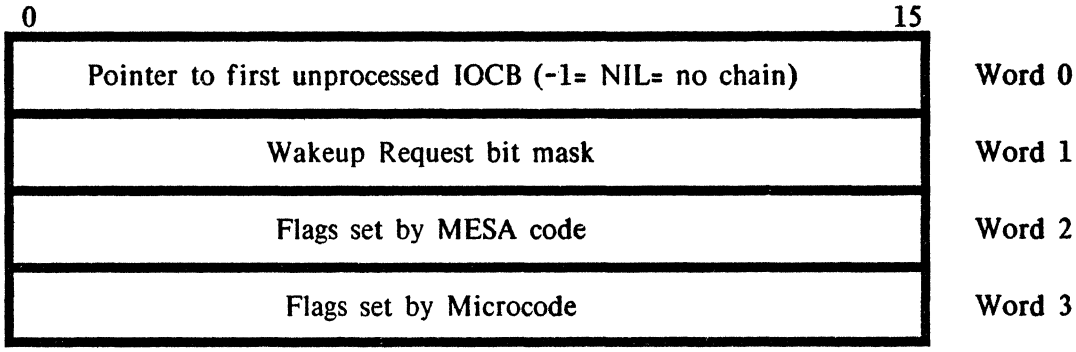
The following parameters relate to the hardware configuration of the D0 and its associated RS232C hardware (including modems and automatic calling equipment). It is assumed that these parameters are available *elsewhere,* and are used in configuring the microcode prior to its initial execution.

modem timeouts, clocking (internal or external), communication facility (half vs. full duplex; switched vs. dedicated), existence of Automatic Calling Equipment (with or without answer-detect circuitry), etc.

Note that the D0 RS232C Controller *will not* provide *receiver signal element timing* (i.e. receive clock) for synchronous lines, but will expect the modem to provide it.

Table Formats

CONTROLLER BLOCK



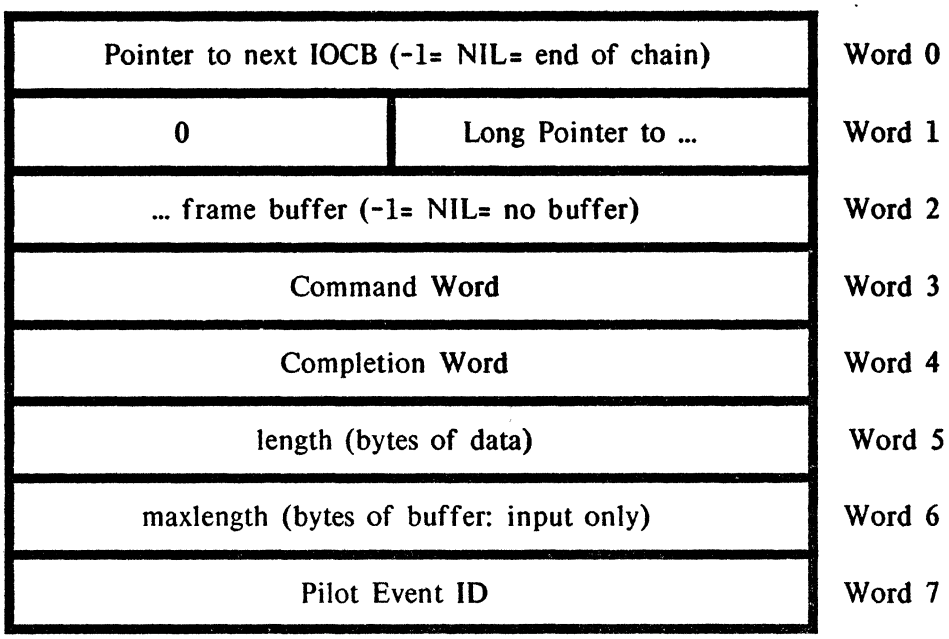
Flags

Bit	Set by MESA code	Set by Microcode
0	Request to Send	Clear to Send
1	Data Terminal Ready	Data Set Ready
2	reserved	Ring Indicator (latched)
3	reserved	Carrier Detect
4	Ring Indicator	reserved
5	reserved	Data lost (latched)
6	reserved	Break detected (latched)
7	reserved	Call Origination Status
8	Call Request	Data Line Occupied
9-14	reserved	reserved
15	Abort	Microcode is running

Note: Input status matches as closely as possible the actual state of the controller. Output status is generated by the controller to match the bits in the Controller Status Block. Latched status bits, when set by microcode, will remain set until cleared by the software (via the *change parameter* command).

DRAFT - DRAFT - DRAFT - DRAFT

IOCB: Reset, Transmit/Receive Frame, Transmit Break Commands



Command Word

bit 0 = Wakeup requested upon completion of IOCB
bit 1 = Wakeup requested if error encountered while processing this IOCB.
bit 2 = Stop if error encountered while processing IOCB; do not proceed to next IOCB.
bit 3 = Start of Frame
bit 4 = End of Frame
bits 5-7 = reserved
bits 8-15= Command:

1= Reset
2= Change Parameter: see IOCB: Change Parameter Command
3= Transmit/Receive Frame
4= Transmit Break: for asynchronous lines only.
5= Dial-a-Digit: see IOCB: Dial-a-Digit Command

Completion Word: 0= IOCB is unprocessed. Else, status word is as follows:

bit 0 = IOCB-has-been-processed flag.
bit 1 = Error
bit 2 = End of Frame
bits 3-7 = reserved
bits 8-15= Status:

1= Success
2= Data lost (overrun)
3= Break detected
4= Frame timeout
5= CRC error
6= LRC/VRC error
7= Invalid character (BSC/ASCII)
8= Block check (BSC)
9= Invalid frame
10= Asynchronous framing error (i.e. stop bit(s) missing)
11= Invalid IOCB (e.g. illegal command or bad parameter)
255= Disaster (e.g. DSR lost): See CSB for further information.

length/maxlength: on output, the software fills the buffer and stores a byte count in *length*; *maxlength* is not used. On input, the software sets *maxlength* to indicate the size of the buffer, and the microcode sets *length* to indicate how many data bytes were stored in the buffer.

DRAFT - DRAFT - DRAFT - DRAFT

IOCB: Change Parameter Command

Pointer to next IOCB (-1= NIL= end of chain)	Word 0
Parameter Descriptor	Word 1
Parameter Mask	Word 2
Command Word	Word 3
Completion Word	Word 4
Parameter Value	Word 5
Pilot Event ID	Word 6

Command Word

bit 0 = Wakeup requested upon completion of IOCB
bit 1 = Stop if error encountered while processing IOCB; do not proceed to next IOCB.
bit 2 = Wakeup requested if error encountered while processing this IOCB.
bits 3-7 = reserved
bits 8-15= Command:

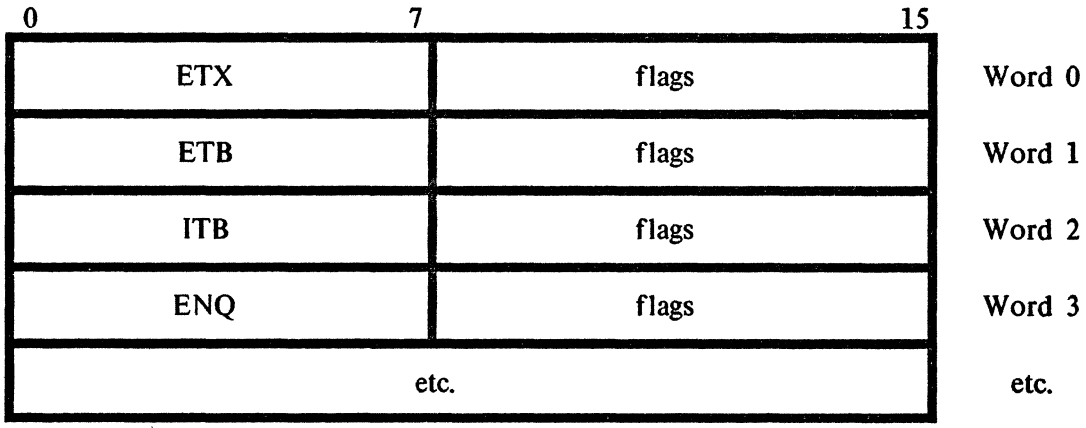
1= Reset: See IOCB: Reset, Transmit/Receive, Transmit Break Commands
2= Change Parameter
3= Transmit/Receive Frame: See IOCB: Reset, Transmit/Receive, Transmit Break Commands
4= Transmit Break: See IOCB: Reset, Transmit/Receive, Transmit Break Commands
5= Dial-a-Digit: see IOCB: Dial-a-Digit Command

Completion Word: See IOCB: Reset, Transmit/Receive, Transmit Break Commands

Parameter descriptor/mask/value: These fields will be set for the convenience of the microcode. It is expected that the descriptor will be a register number, and that the mask will identify the field in the register which is to hold the value. The parameters and legal values to be encoded via this scheme are as follows:

Parameter Name	Parameter Values
Line Speed	110,134.5,150,300,600,1200,2400,3600,4800,7200,9600
Start Bits	N/A, 1
Stop Bits	N/A, 1, 2
Parity	None, Odd, Even, Zero
Character Length	5,6,7,8
Sync. Count	0-n
End-of-Frame Table*	address
Station Address	(to be defined)
Latch-bit-clear mask	-

*End-of-Frame Table: The microcode requires an End-of-Frame Table to determine what constitutes an end of frame (especially for asynchronous input), as well as to take special action when necessary. For bit-oriented link protocols, the concept of frame is sufficiently well defined that no End-of-Frame Table is required. For Asynchronous line protocols, the End-of-Frame Table consists of 256 bits (16 words). If a bit is set, then the 8-bit data byte with the corresponding binary value is to be considered an end-of-frame byte. For BSC line protocol, the End-of-Frame Table looks as follows:

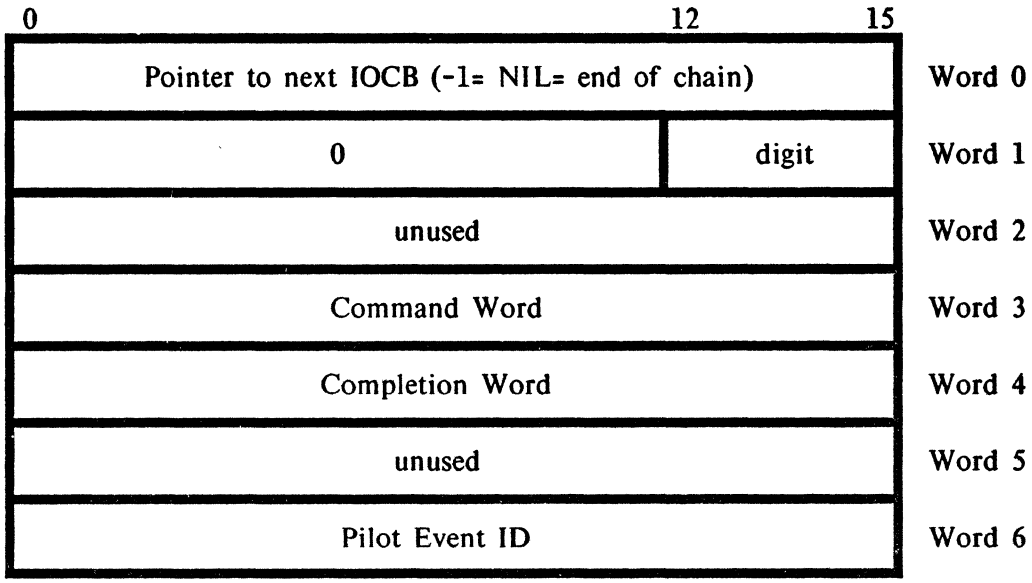


flags

bit 8 = generate/strip Block Check Character (BCC)
bits 9-15 = reserved

DRAFT - DRAFT - DRAFT - DRAFT

IOCB: Dial-a-Digit Command



Command Word

bit 0 = Wakeup requested upon completion of IOCB
bit 1 = Stop if error encountered while processing IOCB; do not proceed to next IOCB.
bit 2 = Wakeup requested if error encountered while processing this IOCB.
bits 3-7 = reserved
bits 8-15= Command:

1= Reset: See IOCB: Reset, Transmit/Receive, Transmit Break Commands
2= Change Parameter: see IOCB: Change Parameter Command
3= Transmit/Receive Frame: See IOCB: Reset, Transmit/Receive, Transmit Break Commands
4= Transmit Break: See IOCB: Reset, Transmit/Receive, Transmit Break Commands
5= Dial-a-Digit: Binary digit is in word 2

Completion Word: See IOCB: Reset, Transmit/Receive, Transmit Break Commands

Frame Formats

A frame, as supplied or received by the software, consists of an integral number of 8-bit bytes in the code set expected by the correspondent. Note that certain elements of the frame, as it travels across the communication line, are generated/stripped by the microcode as described below. Hence the format of a frame at the interface between the software and the microcode is slightly different from the frame format as shown in the corresponding protocol documentation. The 8-bit bytes are serialized across the RS232C interface with the following transformations performed by the controller:

Bit-Oriented (HDLC, SDLC, ADCCP): Flag patterns (01111110), Frame Check Sequences (FCS) Station Address, and synchronization information are generated (output) and checked/stripped (input) by the controller for all frames. Zero insertion/removal following "11111" patterns is performed for all frames. On input, end-of-frame is defined by the recognition of a second Flag pattern. On output, end-of-frame is defined by the software setting the end-of-frame flag in the IOCB.

BSC: VRC, LRC, Station Address, and synchronization information are generated (output) and checked/stripped (input) by the controller. End-of-frame is defined by recognition of a byte value contained in the End-of-Frame Table (see Change Parameter Command) and the controller generates or checks the BCC according to a flag in that table.

Asynchronous: Parity and start/stop bits are generated (output) and checked/stripped (input) by the controller for each byte. On input, end-of-frame is defined by the recognition of a byte value which is flagged in the End-of-Frame Table (see Change Parameter Command). On output, end-of-frame is defined by the software setting the end-of-frame flag in the IOCB.

c: SDD/SD/CS
Linda Bergsteinsson
Jozef Furst
Peter Heinrich
Pitts Jarvis
David Liddle
Ruben Loshak
Bill Lynch
Ed Miller
Brian Rosen
Wendell Shultz
Dan Stottlemeyer
Jerry Szelong
Chuck Thacker