XEROX

High-end IOT Interface 2.2

Xerox Private Data

Xerox System Integration Guide

High-end IOT Interface 2.2 Xerox System Integration Guide XNSGS 108911

XEROX

High–end IOT Interface 2.2

XEROX
Private
Data

November 1989
XNSGS 108911

Xerox Corporation
Printing Systems Division
Printing Systems Administration Office
701 South Aviation Blvd.
El Segundo, California 90245

November 1989

Publication #XNSGS 108911

Printed in the United States of America

Changes are periodically made to this document. Changes, technical inaccuracies, and typographic errors will be corrected in subsequent editions.

This document was created on the Xerox 6085 workstation using ViewPoint software. The typeface is Optima.

Preface

This document is one of a set of Xerox internal documents which prescribe the interfaces between Print Service Processors and Image Output Terminals of high-end and low-end electronic printers. It is not a Xerox Standard, but rather a statement of generic specifications for these interfaces.

High-end IOT interface 2.2 supersedes High-end IOT interface 2.1, XNSGS 108702, February 1987 plus Addendum 1 as well as the OEM version, XNSGS 118702. There is no longer an OEM version. Version 2.2 includes all of the changes contained in Addendum 1, plus a number of changes which were requested subsequent to Addendum 1. All of these changes are to chapter 6, "Client layer protocol." In addition, the handling of highlight color and grey scale are standardized in a new appendix E. There are no physical changes to the interface.

Comments and suggestions on this document and its use are encouraged. Please address communications to:

Xerox Corporation
Printing Systems Division
Printing Systems Administration Office
701 South Aviation Blvd.
El Segundo, California 90245

Table of contents

List of figures

1. Scope

This generic specification prescribes the required interface between the Print Service Processor (PSP) and the Image Output Terminal (IOT) of all *high-end* electronic printers, including those of the Distributed Printing and Reprographics Systems (DPRS) architecture. Refer to figure 1-1. For purposes of this specification, *high-end* includes any printer whose serial video data rate is equal to, or greater than, 10 Megabits per second and whose speed or functionality requires the synchronous data link prescribed herein. (In the remainder of this specification, it is to be understood that all references to PSPs, IOTs, and printers imply *high-end*. Also, the subject interface will be referred to simply as *the Interface* when there is no likelihood of confusion.)

This specification is applicable to all IOTs which print bit-map formats by means of a Raster Output Scanner (ROS) or by means of a line-marking array. This specification is not applicable to printing terminals which mark by means of optically or mechanically pre-formed characters.

This specification accommodates video data transfer rates at the Interface up to 25 Megabits per second serial, and up to 200 Megabits per second parallel.

This specification defines the interchange of video data (bit-map data), command/status data, timing signals, and control signals, between one PSP and one IOT constituting a high-end electronic printer.

This specification defines the layers of a three-layered architecture for command/status communications.

This specification prescribes the interface signal repertoire, including the functional description of all the signals, the signal formats, and the timing relationships, where applicable.

This specification prescribes the electrical and mechanical characteristics and parameters of the signal circuits, as well as the control procedures for effecting the transfer of information across this interface.

This specification prescribes the interface circuit design and the required electronic and mechanical components (cables and connectors).

This specification specifies grounding and shielding requirements associated with the interface.

This specification defines the interface compatibility required between PSPs and IOTs of different performance capabilities.

For purposes of this specification, an IOT is defined in terms of its functional entities as follows.

An IOT includes:

- Marking engine
- Raster output scanner (ROS) or line marking array
- Paper transport and delivery system
- Machine controller(s)
- Electronic interface to Print Service Processor
- DC power supply(s)
- AC power interface and controls.

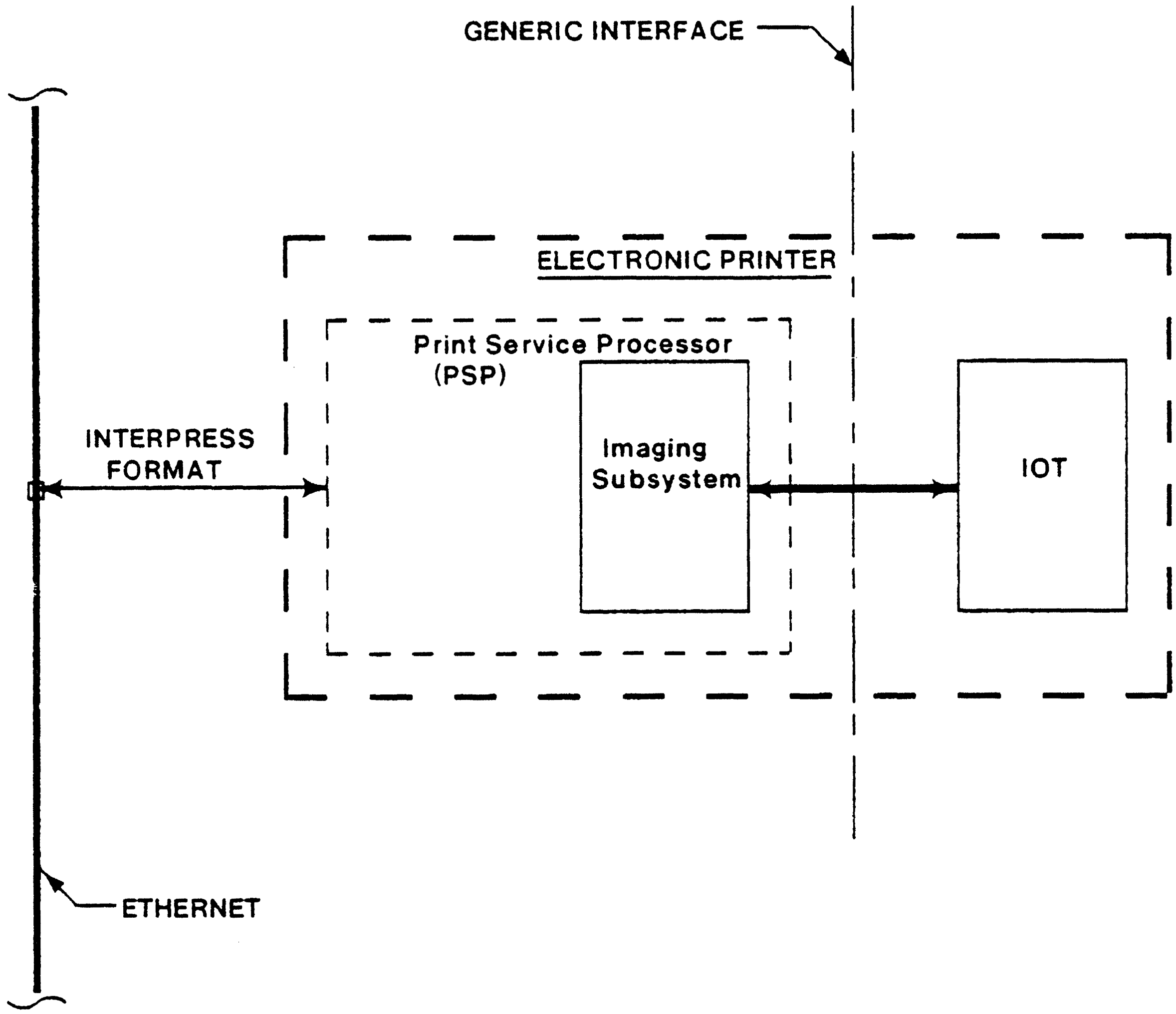
An IOT may include:

- User interface
- Maintenance interface
- Video data buffer
- Command buffer
- Test pattern generator
- Stand-alone diagnostics
- Light-lens platen and optics
- Document handler for light-lens copying
- Stacker
- Sorter
- Billing meter(s).

An IOT does not include:

- Video generator, (character dispatcher/image generator)
- Font storage
- Data decompressor
- Application processors
- File storage facilities
- Format processor or format-standard level shifter
- Standard system interface (Ethernet).

Figure 1-1. **System location of generic interface**



2. Introduction

The purpose of this specification is to provide a common implementation for the hardware and software interconnection between PSPs and IOTs of electronic printers. It is intended to be general enough to accommodate IOTs with different performance capabilities, different line scanning formats, and different marking technologies. This specification requires that each PSP be compatible, at this interface, with all IOTs of performance equal to or less than its peer IOT, within the bounds of *high-end* performance (chapter 1). This specification is intended to be sufficiently specific and complete so as to allow such interconnection with *no significant* hardware or software changes related to the interface functions. However, each PSP and each IOT will have some unique parameters which must be made known to each other. This specification identifies these parameters, but does not prescribe the method by which they are introduced into the system.

The IOT Interface is the path by which bit-maps of the images to be printed reach the IOT. The interface embodies the communications by which the electronic image generation functions of the PSP are synchronized with the real-time requirements of the marking engine, and by which the PSP controls the operational states of the IOT. The command/status interface also provides the means for transferring blocks of non-image data between PSP and IOT (and vice-versa), and for invoking and/or running IOT diagnostic routines from the PSP.

The transfer of an image is viewed as a real-time process between PSP and IOT, with the IOT as the source of synchronization for line and page generation. The fast scan and/or slow scan mechanisms of the marking engine may or may not be decoupled in time from the interface transfer. In any case, the IOT defines a *standard image frame*, and it is the responsibility of the IOT to align this frame relative to the elements of the marking engine and to the output paper. It is the responsibility of the PSP to align the image within the standard image frame. Control of marking outside of the standard image frame is an IOT responsibility and is not supported at the interface.

The primary format for video data at the interface is 8-bit parallel; however, a serial mode is also provided for video data rates under 25 Megabits per second. The PSP provides a return video clock in both cases. In the serial mode, the use of the return video clock in the IOT is optional.

Application of this Generic Interface Specification to some complex printing arrangements which have been proposed may not be feasible because of the combined constraints imposed on data rate, grounding, and shielding.

The PSP and IOT of a printer may be co-packaged with minimal separation in a single cabinet, or may be contained in physically separated individual cabinets. This specification prescribes the interconnecting cable to be used, includes a timing tolerance analysis for the reclocked video data transmission modes, and specifies jitter limits for the unreclocked mode.

3. Interface signals functional description

Note: In the following text, boldface is used to indicate the actual signal as it appears at the interface, as distinguished from use of the term as an adjective or in reference to the function performed.

3.1 Imaging interface

3.1.1 Imaging conventions

The imaging interface signals are described with respect to the following spatial imaging conventions, which must be followed by all PSPs and IOTs. The *standard image frame* is a rectangular image space whose sides are parallel to the slow scan and fast scan directions of the IOT, and whose dimensions are equal to the dimensions of the largest size paper which an IOT can accommodate, plus registration tolerance in the respective directions. Figure 3-1 shows the generalized relationships when viewing the imaged side of the paper. In the slow scan direction, the total paper registration error (plus and minus) comprises the registration tolerance. In the fast scan direction, there is an additional image generation tolerance, specified in chapter 4, to allow for priming and flushing the video processing channels in the PSP. Each IOT design must specify its own unique standard image frame(s). The first pixel delivered to the standard image frame must fall at the lower left hand corner, as viewed from the start-of-scan edge, and the fast scan and slow scan must proceed upward and to the right, respectively. (Although long-edge feed is shown in figure 3-1, no inference should be made that this is a requirement.)

It is the responsibility of the IOT to align the elements of the marking engine so that the maximum size paper is registered in the center of the standard image frame in both directions, within the limits of the registration errors. It is the further responsibility of the IOT to register smaller size paper according to one of the modes listed below, which must be known to the PSP. *Leading* and *trailing* refer to the slow scan direction. *Start-of-scan, end-of-scan,* and *center* refer to the fast scan direction.

PAPER REGISTRATION MODES

A. Leading-edge, Start-of-scan
B. Leading-edge, Center
C. Leading-edge, End-of-scan
D. Trailing-edge, Start-of-scan
E. Trailing-edge, Center
F. Trailing-edge, End-of-scan

Registration of the respective edges (or center) is with respect to the ideal location of the corresponding edges (or center) of the maximum size paper. Note that with gapless printing, there are no paper registration errors in the slow scan direction.

Page video is that image information which is intended to appear on paper. It is the responsibility of the PSP to align the page video within the standard image frame so that it is imaged on paper for any combination of paper size and registration mode. The PSP may offset the image forward or backward in the slow scan direction so that some of the original page video, at the leading or trailing edge, is not actually transmitted to the IOT and, thus, is not imaged. If a PSP cannot offset images in this manner, the desired result may be obtained by requesting that the IOT define its standard image frame to be larger in the slow scan direction by the total amount of leading and trailing offset desired. In this case, the page video in the offset area is actually transmitted to the IOT, but is imaged beyond the leading or trailing edge of the paper.

When edge marking is employed, the PSP may designate page video to the outer limits of the expected position of the paper, i.e., including the registration errors. This is done to guarantee that the desired image falls at the extreme edge of the paper. If the PSP's fast scan image registration tolerance is such as to preclude reaching the edge of paper (at either end), IOT registration adjustment of the standard image frame can achieve the desired result. With edge marking, it is likely that some page video will fall beyond the edge of the paper.

Note that an IOT may implement more than one Standard Image Frame. The procedure for changing from one to the other is IOT specific. See also application notes for information type, "MediaMatrix," under IOT status message, "IotConfiguration," in section 6.3.3.

3.1.2 Page sync

This signal is generated in the IOT and sent to the PSP. It defines each slow scan imaging interval to the PSP in real time. Its primary purpose is to synchronize the delivery of a standard image frame worth of video data to the IOT. The PSP must deliver a standard image frame worth of video data for each **page sync** received. The duration of **page sync** defines the standard image frame in the slow scan direction. The beginning of **page sync** must correspond to the leading edge of the standard image frame, and the end of **page sync** must correspond to the trailing edge of the standard image frame. The PSP must be capable of delivering scan lines containing page video starting with the first line sync following the beginning of each page sync. This capability is required to accomplish edge marking at the leading edge of paper, when the paper is leading edge registered.

An exception to the above definition is necessary for gapless printing. In this case, **page sync** must end prior to the end of the standard image frame currently being delivered by the PSP. The PSP must continue to deliver scan lines for the current standard image frame until the beginning of the next **page sync** occurs, and must then deliver the first scan line of the next standard image frame with the next **line sync**. For this purpose, the PSP must know that the IOT is a gapless printer.

(Section 6.4.2, "Message sequencing and timing requirements," defines the relationship between certain command/status messages and page sync. Section 6.4.4, "Page sync regimen," discusses the rules for appearance of page sync at the physical interface.)

3.1.3 Line sync, video clocks, video data

To accommodate the anticipated speed and functional requirements of the various electronic printer products, two different interface modes are prescribed for this set of signals, as follows.

3.1.3.1 Eight-bit parallel mode (refer to figure 3-2)

This is the principle interface mode, designed to accommodate total video data rates up to 200 Mbps. In this mode, the video clock is a byte clock with maximum frequency of 25 Mhz. This clock is returned to the IOT, along with the parallel video data clocked out of the PSP, and is used to reclock this parallel video data at the IOT before it is used for marking. In order to achieve the desired degree of compatibility across the product line, all PSPs which implement this parallel interface must also support the Serial Mode at 25 Mbps (refer to section 3.1.3.2).

Line sync This signal is generated in the IOT and sent to the PSP. It defines each fast scan imaging interval to the PSP in real time. It helps to synchronize the delivery of a scan line worth of video data to the IOT. The PSP should re-synchronize on every **line sync**, so as not to propagate the effects of any loss of synchronism. **Line sync** must be present during every page sync and may be present during interpage gaps and during other non-imaging states of the IOT. The positive transition of **line sync** marks the nominal start of the standard image frame in the fast scan direction (the exact start is defined below). The negative transition of **line sync** marks the end of the standard image frame.

Byte clock This signal is generated in the IOT and sent to the PSP. It defines the occurrence and time span of each byte interval during which eight pixels are transmitted in parallel from the PSP to the IOT. **Byte clock** occurs in bursts, beginning a specified time after the beginning of **line sync**, and ending with the falling edge of line sync. (An allowable overrun into line sync dead-time is defined in section 4.2.1.) The beginning of the byte clock burst marks the beginning of the standard image frame in the fast scan direction. **Byte clock** must be transmitted whenever **line sync** is transmitted. The quiescent state of byte clock is logic level 0. The average period of the byte clock cycles during a line sync period may vary slightly from scan to scan, according to the design requirements of the IOT.

Return byte clock This signal is derived from **byte clock** in the PSP and is transmitted from the PSP back to the IOT, along with **video data**. It is intended that return byte clock should undergo the same propagation delay as video data, so as to be directly usable for reclocking **video data** at the IOT. (Refer to section 7.2 for analysis of timing tolerances.)

Video data In this mode, there are eight parallel video data signals designated **video data 0** through **video data 7**. These signals are generated in the PSP using **byte clock**, and sent to the IOT. The earliest through latest pixels of an 8-pixel octet of scan line data must be placed on video data lines **video data 7** through **video data 0**, respectively. The full content of the scan-line (defined by the standard image frame) must be delivered by the PSP. It is the responsibility of the PSP to align the page video within the standard image frame so as to account for the current paper size and the paper registration mode being used by the IOT. For all pixel positions which precede or follow page video within the standard image frame, the PSP must deliver pixels with the image background level. Signal lines video data 0 through video data 7 must also be placed at the logic level corresponding to image background, during all line sync dead times and during all page sync dead times (except with gapless printing, refer to section 3.1.2), including machine states when **page sync** is inactive and including machine states when the video generator is inactive. It is the responsibility of the IOT to manage the state of the marking means outside of the standard image frame.

An entire standard image frame of background level video will be delivered when a *dead cycle* is requested at the Client Layer (refer to chapter 6).

3.1.3.2 Serial mode, reclocked and un-reclocked (refer to figure 3-3)

When the total video data rate at the Interface is not greater than 25 Mbps, the serial interface may be used. In this mode, video data is clocked out of the PSP, transmitted serially to the IOT, and either used directly for marking without reclocking, or reclocked at the IOT for functional or signal quality reasons before marking. In this mode, the video clock is a pixel clock.

Line sync This signal is generated in the IOT and sent to the PSP. It defines each fast scan imaging interval to the PSP in real time. It helps to synchronize the delivery of a scan line worth of video data to the IOT. The PSP should re-synchronize on every line sync, so as not to propagate the effects of any loss of synchronism. Line sync must be present during every page sync and may be present during interpage gaps and during other non-imaging states of the IOT. The positive transition of line sync marks the nominal start of the standard image frame in the fast scan direction (the exact start is defined below). The negative transition of line sync marks the end of the standard image frame.

Pixel clock This signal is generated in the IOT and sent to the PSP. It defines the occurrence and time-span of each pixel as it is intended to be received by the IOT. It is used to clock the video data out of the PSP for transmission across the Interface. **Pixel clock** occurs in bursts, beginning a specified time after the significant transition of line sync, and ending at the falling edge of line sync (an allowable overrun into line sync dead-time is defined in section 4.2.2). The beginning of the pixel clock burst marks the beginning of the standard image frame in the fast scan direction. The quiescent state of pixel clock is logic level 0. The average period of the pixel clock cycles during a line sync period may vary from scan to scan, according to the design requirements of the IOT.

Return pixel clock This signal is derived from **pixel clock** in the PSP and is transmitted from the PSP back to the IOT along with **video data**. It is intended that **return pixel clock** should undergo the same propagation delay as **video data** so as to be directly usable for reclocking **video data** at the IOT (refer to section 7.2 for analysis of timing tolerances). For purposes of product line compatability all PSPs which implement the serial mode must implement **return pixel clock**. IOTs may or may not use **return pixel clock**, depending on their design requirements.

Video data This signal is generated in the PSP and sent to the IOT. It is generated in the PSP using **pixel clock**. The full content of the scan line (defined by the standard image frame) must be delivered by the PSP. It is the responsibility of the PSP to align the page video within the standard image frame, so as to account for the current paper size and the paper registration mode being used by the IOT. For all pixel positions which precede or follow page video within the standard image frame, the PSP must deliver pixels with the image background level. The video data signal line must also be placed at the logic level corresponding to image background during all line sync dead times, and during all page sync dead times (except with gapless printing), including machine states when **page sync** is inactive and including machine states when the video generator is inactive. It is the responsibility of the IOT to manage the state of the marking means outside of the standard image frame. When a PSP with a parallel interface is used in the serial mode, the video signal must be placed on the **video data 7** interface path.

An entire standard image frame of background level video will be delivered when a *dead cycle* is requested at the client layer (refer to chapter 6).

3.2 Control interface

3.2.1 Command

Command is generated in the PSP and sent to the IOT in the physical layer of the multi-layer command/status function (refer to chapter 5). This is a message formatted signal carrying command information to the IOT, necessary for managing the printer in its various running modes. It is also suitable for downloading software from the PSP to the IOT. The transmission protocol is HDLC (High-level Data Link Control protocol). This signal must be encoded so as to allow clock recovery from the serial data stream at the receiver.

3.2.2 Status

Status is generated in the IOT and sent to the PSP in the physical layer of the multi-layer command/status function (refer to chapter 5). This is a message formatted signal which carries status information from the IOT, necessary for management of the printer in its various running modes. It is also suitable for uploading blocks of information from IOT memory, upon command from the PSP. The transmission protocol and encoding requirements are the same as for Command.

3.2.3 Power control

This is a special circuit from the PSP to the IOT, which can cause the main AC power in the IOT to be turned on or turned off, provided that the IOT has been placed in a "Remote Power Control" mode. (The IOT must have an Emergency Power Off switch which can override this remote power control.) This circuit provides DC power from a PSP power source to a control actuator in the IOT, and includes a return path to the PSP in which the primary control elements are located. The control actuator in the IOT operates a power actuator which switches the AC power mains in the IOT (refer to section 8.4).

The power on/off condition of the IOT may be monitored by detecting in the PSP whether or not the status signal source has its normal DC power (refer to section 8.5). There is no separate interface circuit to monitor the power on/off status of the IOT.

It is the individual responsibility of the PSP and the IOT to manage their own shutdown modes. However, advance warning of a non-emergency shut-down initiated by the PSP must be sent to the IOT (refer to chapter 6, PspMetaReset and IotConfiguration).

3.3 Alternate imaging modes

Conventions for handling trilevel highlight color imaging and grey scale imaging are prescribed in appendix E.

Figure 3-1. **Spatial relationships for imaging (viewing imaged side of paper)**

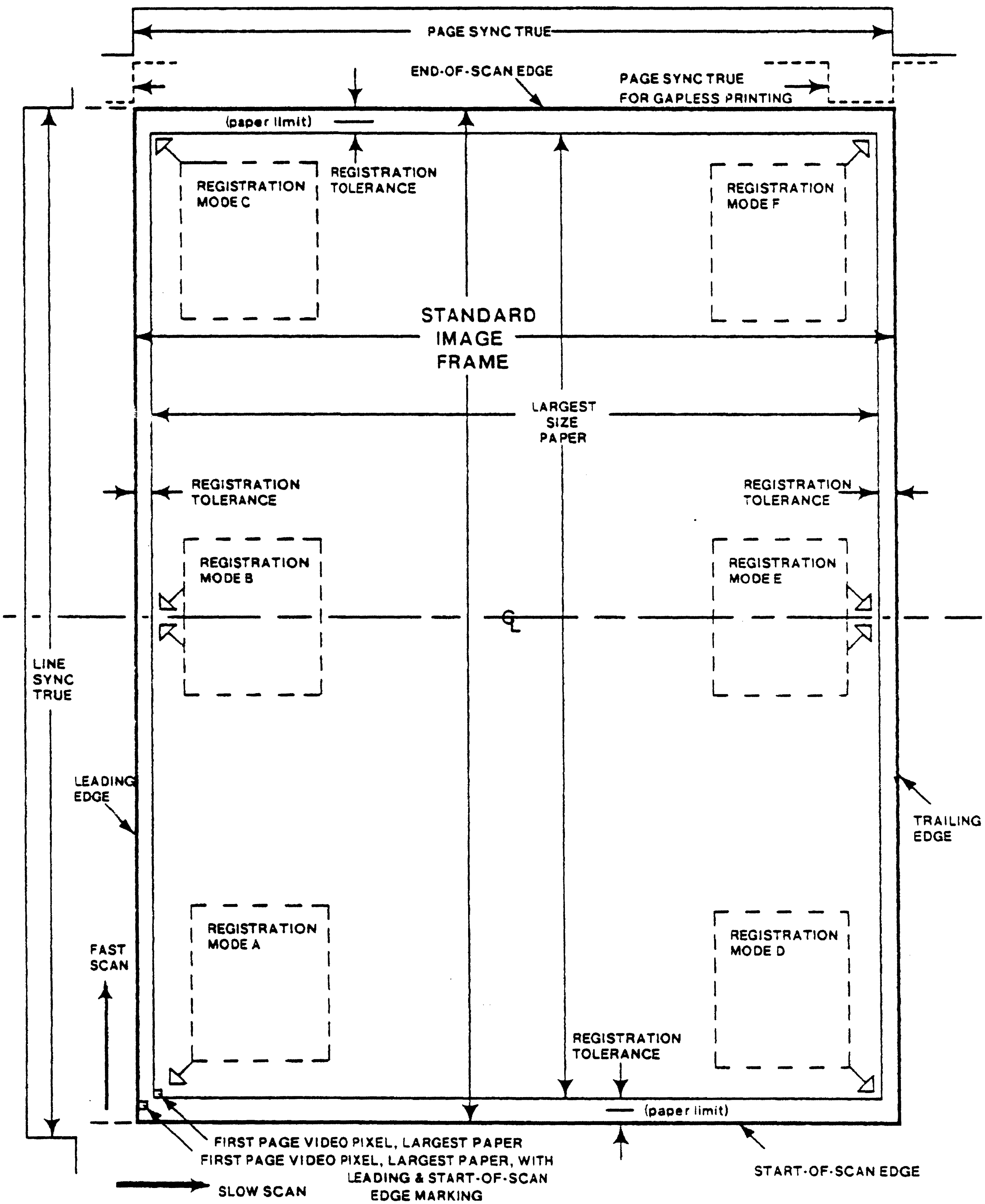


Figure 3-2. **Summary of generic interface signals for 8-bit parallel video data transmission mode**

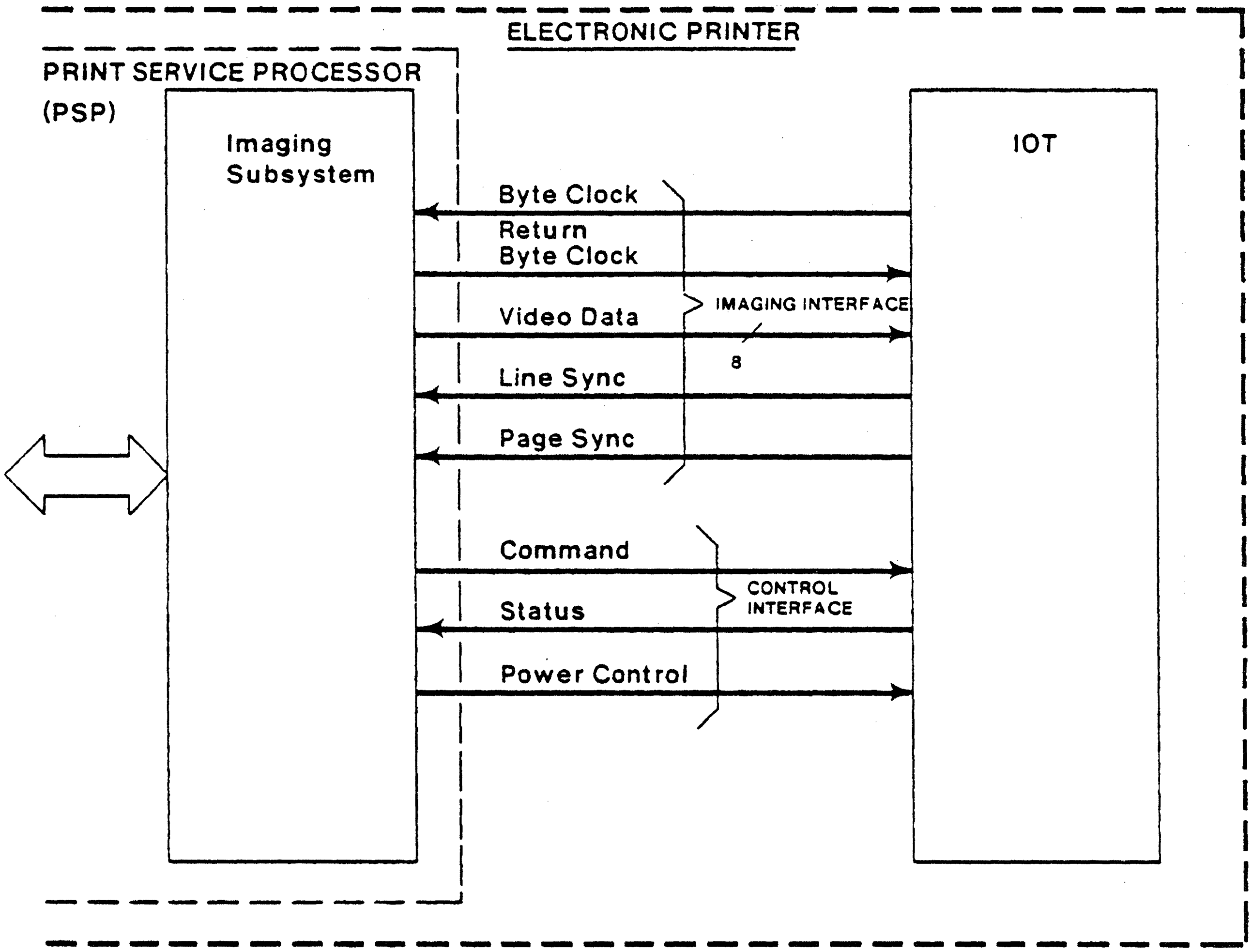
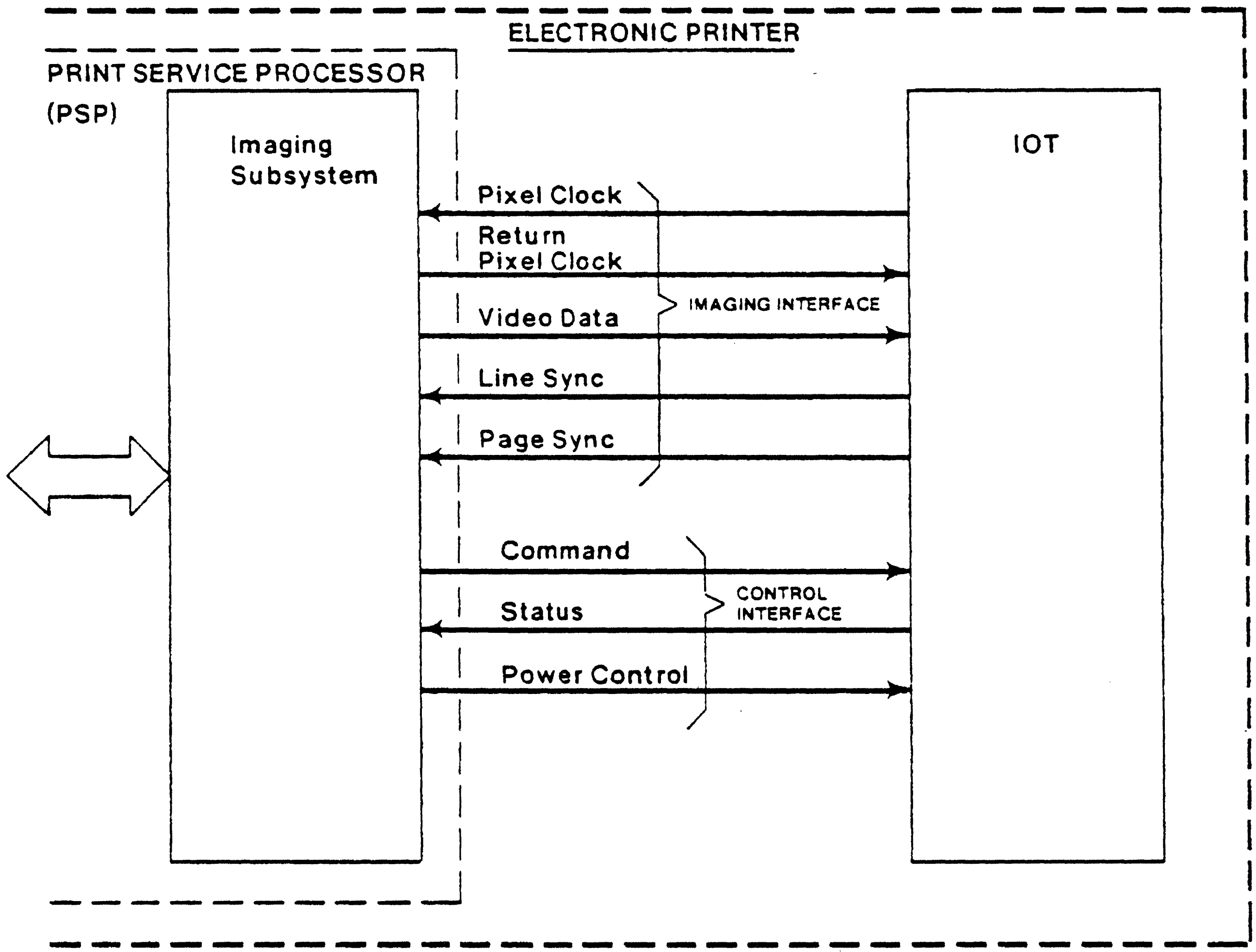


Figure 3-3. **Summary of generic interface signals for serial video data transmission mode**



4. Characteristics of imaging interface signals

The following nomenclature conventions have been adopted for the Imaging Interface signals described below:

Signal Name +	A single rail of the differential signal; high=logic 1
Signal Name –	A single rail of the differential signal; the complement of Signal Name +
Logic 1	The most positive signal voltage level: typically –0.8v when referenced to signal ground (logic ground)
Logic 0	The most negative signal voltage level: typically –1.8v when referenced to signal ground (logic ground).

4.1 Page sync (refer to figure 4-1)

The principal description of page sync which follows applies to IOTs which utilize cut sheet paper. Gapless printers which utilize a continuous paper web are accommodated by exception, also described below.

Page sync is a binary signal whose true state delineates the slow scan dimension of the standard image frame to the PSP. Thus, the rising edge (positive transition) of **page sync** indicates to the PSP that it must start delivering the first scan line of the standard image frame. Positive transitions are from the 0 logic level to the 1 logic level, as observed on the *positive rail* of the balanced interface connection (the *negative rail* will have the logical inverse transitions). **Page sync** is considered to be *true* when it is in the logic 1 state, as defined above. Both the rising and falling edges of **page sync** must be synchronized to the falling edges of **line sync** with a tolerance of ±50 nanoseconds at the IOT end of the interface cable. This will guarantee that the first rising edge of **line sync** following the rising edge of **page sync** can be reliably detected by the PSP after suffering propagation skew. It also defines the minimum page sync dead-time to be equivalent to one line sync interval. **Page sync** may recur regularly or irregularly (refer to section 6.4.4, "Page Sync Regimen," and the operating sequences in section 6.5). In either case, the IOT design must specify the minimum *page time*, i.e., the minimum time between any two successive rising transitions of **page sync**. The quiescent state of **page sync** is the 0 logic level on the positive rail of the balanced interface connection.

For gapless printing IOTs the standard image frames are contiguous and the rising edges of **page sync** mark the boundries in the slow scan direction. Therefore, the falling edge of **page sync** must occur within the standard image frame (refer to figure 4-2). The minimum dead time interval must be the same as that for the cut sheet case. A PSP must know when it is connected to a gapless printing IOT, and, in that case, must continue to deliver standard image frame scan lines during **page sync** false, until the end of the standard image frame is reached. This is determined either by line count or by occurrence of the next positive transition of page sync. The first scan line of the next standard image frame must be delivered with the first positive transition of **line sync** following the page sync transition. The PSP must notify the IOT of what slow scan dimension is appropriate for the standard image frame.

4.2 Line sync, video clocks, video data

4.2.1 Eight-bit parallel mode (refer to figure 4-3)

Line sync This is the primary timing reference for this group of signals. **Line sync** is a binary signal whose true state delineates the fast scan dimension of the standard image frame to the PSP. Thus, the rising edge (positive transition) of **line sync** indicates to the PSP that it must prepare to deliver the first byte of the standard image frame. Positive transitions are from the 0 logic level to the 1 logic level, as observed on the positive rail of the balanced interface connection (the negative rail will have the logical inverse transitions). **Line sync** is considered to be true when it is in the logic 1 state, as defined above. The actual duration of line sync true must include all of the byte clocks which constitute the standard image frame, plus a minimum of 120 nanoseconds preceding the first transition of **byte clock** (refer to the next section). **Line sync** may recur regularly or irregularly. The dead-time interval must be a minimum of 300 nanoseconds, so as to provide a margin at the IOT between the end of **return byte clock** and the beginning of byte clock for the next scan line. The quiescent state of **line sync** is the 0 logic level on the positive rail of the balanced interface circuit.

Byte clock This is a long burst of square waves whose positive transitions are used to clock the video data out of the PSP eight pixels at a time in parallel. The clock period, T_{BC}, must be specified by the IOT and must not be less than 40 nanoseconds. It will normally be derived from an internal IOT pixel clock. The duty cycle must be 50% $\pm$5% at the input to the line driver in the IOT. Rising edges (positive transitions) are from the 0 logic level to the 1 logic level, as observed on the *positive rail* of the balanced interface connection. The delay between the positive transition of **line sync** and the first positive transition of **byte clock** must not be less than 120 nanoseconds at the input to the line driver at the IOT interface, so as to guarantee at least 100 nanoseconds delay at the output of the line receivers at the PSP interface. The number of cycles of **byte clock** which constitute a scan line of the standard image frame is equal to the number of 8-pixel octets of video data, which represent the largest size paper that the IOT will accommodate plus the IOT's paper registration tolerance, plus an additional 24 to 48 byte clocks for PSP image registration tolerance. The PSP may be designed to divide the extra 24 to 48 byte clocks between the beginning and end of scan, as needed. This division is to be fixed for any given PSP design and not alterable from line to line or from page to page. If line sync dead-time is greater than 300 nanoseconds, an IOT may continue byte clocks into the dead-time interval of **line sync**, but byte clock cycles must terminate at least 300 nanoseconds before the next rising edge of **line sync**. Fractional byte clocks are not allowed. The true state must always be a full half cycle. The quiescent state of byte clock is logic level 0 on the positive rail of the balanced interface circuit.

Return byte clock This is a binary signal derived from **byte clock** at the PSP. At the input to the interface line drivers in the PSP, it must be phased so that the positive transitions are centered in the video data byte interval with a tolerance of ± 5 percent. This will guarantee that **video data** can be reliably reclocked at the output of the line receivers in the IOT. Positive transitions are from the 0 logic level to the 1 logic level, as observed on the *positive rail* of the balanced interface connection. **Return byte clock** must have essentially the same epoch as **byte clock**, and exactly the same number of cycles.

Video data These are eight binary signals, each of which follows the same polarity convention; logic level 0 on the positive rail of the balanced interface (logic level 1 on the negative rail) corresponds to image background level. The nominal time period of each byte of video data is determined by the period of **byte clock**, as received at the PSP. All of the **video data** signals must be in phase with each other (and with **return byte clock**, as described above) at the input to the line drivers at the PSP interface. The timing relationship between r**eturn byte clock** and each of the **video data** signals at the output of the line receivers in the IOT depends on the propagation delay tolerances of the interface circuits and cables of the respective paths (refer to section 7.2 for an analysis of the timing tolerances). The absolute delay of **video data** relative to line sync or **byte clock** *at the IOT* depends on the cable lengths employed in the interface connection, and is not critical. The procedures for image registration in the fast scan direction will automatically compensate for this delay. The number of bytes of page video per scan line will be equal to integer N/8, where N is the number of pixels per scan line of the current image. The PSP must align the page video pixels within the standard image frame, in both the slow and fast scan directions, according to the paper registration mode of the IOT and the current paper size. The accuracy of this alignment is a PSP performance parameter, not specified in this standard.

The PSP must guarantee that video data on all lines corresponds to the background level during the registration and overscan byte clocks, and must also place all video data lines at the background level whenever byte clocks are not present.

4.2.2 Serial mode (refer to figure 4-4)

Line sync This is the primary timing reference for this group of signals. **Line sync** is a binary signal whose true state delineates the fast scan dimension of the standard image frame to the PSP. Thus, the rising edge (positive transition) of **line sync** indicates to the PSP that it must prepare to deliver the first pixel of the standard image frame. Positive transitions are from the 0 logic level to the 1 logic level, as observed on the *positive rail* of the balanced interface connection (the *negative rail* will have the logical inverse transitions). **Line sync** is considered to be true when it is in the logic 1 state, as defined above. The actual duration of line sync true must include all of the pixel clocks which constitute the standard image frame, plus a minimum of 120 nanoseconds preceding the first transition of **pixel clock** (refer to the following section). **Line sync** may recur regularly or irregularly. The dead-time interval must be a minimum of 300 nanoseconds so as to provide a margin at the IOT between the end of **return pixel clock** and the beginning of pixel clock for the next scan line. The quiescent state of **line sync** is the 0 logic level on the positive rail of the balanced interface circuit.

Pixel clock This is a long burst of square waves whose positive transitions are used to serially clock the video data out of the PSP. The clock period, T_{PC}, must be specified by the IOT and must not be less than 40 nanoseconds. The duty cycle of **pixel clock** is nominally 50 percent. For IOTs which reclock **video data**, the duty cycle tolerance is ±5 percent. This tolerance requirement is applicable at the input to the line driver in the IOT. Rising edges (positive transitions) are from the 0 logic level to the 1 logic level, as observed on the *positive rail* of the balanced interface connection. The delay between the positive transition of **line sync** and the first positive transition of **pixel clock** must not be less than 120 nanoseconds at the input to the line driver at the IOT interface, so as to guarantee at least 100 nanoseconds delay at the output of the line receiver at the PSP interface. The number of cycles of **pixel clock** which constitute a scan line of the standard image frame is equal to the number of pixels of video data which represent the largest size paper that the IOT will accommodate, plus the IOT's paper registration tolerance, plus an additional 192 to 384 pixel clocks (24 to 48 byte clocks equivalent) for PSP image registration tolerance. The PSP may be designed to divide the extra 192 to 384 pixel clocks between the beginning and end of scan, as needed. This division is to be fixed for any given PSP design and not alterable from line to line or from page to page. If line sync dead-time is greater than 300 nanoseconds, an IOT may continue pixel clocks into the dead-time interval of **line sync**, but pixel clock cycles must terminate at least 300 nanoseconds before the next rising edge of **line sync**. Fractional pixel clocks are not allowed. The true state must always be a full half cycle. The quiescent state of pixel clock is logic level 0 on the positive rail of the balanced interface circuit.

Return pixel clock This is a binary signal derived from **pixel clock** at the PSP. At the input to the interface line drivers in the PSP, it must be phased so that the positive transitions are centered in the video data pixel interval with a tolerance of ±5 percent. This will guarantee that **video data** can be reliably reclocked at the output of the line receiver in the IOT. Positive transitions are from the 0 logic level to the 1 logic level, as observed on the *positive rail* of the balanced interface connection. **Return pixel clock** must have essentially the same epoch as **pixel clock**, and exactly the same number of cycles.

Video data This is a binary signal with polarity convention as follows: logic 0 level on the *positive rail* (logic 1 level on the *negative rail*) of the balanced interface corresponds to image background level. The nominal time period of a pixel of video data is determined by the period of pixel clock as received at the PSP. At the input to the line drivers at the PSP interface, video data must be phased relative to **return pixel clock**, as described previously. The timing relationship between **return pixel clock** and **video data** at the output of the line receivers in the IOT depends on the propagation delay tolerances of the interface circuits and cables of the respective paths (refer to section 7.2 for an analysis of the timing tolerances). The absolute delay of **video data** relative to **line sync** or **pixel clock** *at the IOT* depends on the cable lengths employed in the interface connection and is not critical. The procedures for image registration in the fast scan direction will automatically compensate for this delay. The number of page video pixels per scan line will be equal to the number of pixels per scan line of the current image. The PSP must align the page video pixels within the standard image frame, in both the slow and fast scan directions, according to the paper registration mode of the IOT and the current paper size. The accuracy of this alignment is a PSP performance parameter, not specified in this standard.

The PSP must guarantee that **video data** corresponds to the background level during the registration and overscan pixel clocks, and must also place the video data line at the background level whenever pixel clocks are not present.

When **video data** is not reclocked at the IOT, the jitter of the data transitions caused by the interface transmission will appear as corresponding spatial displacements of the pixel edges on the output paper (refer to chapter 7). For the worst case conditions allowed under this standard, i.e., 25 Megabits per second over 50 feet of cable, the jitter will be well under 5 percent.

Figure 4-1. **Slow scan timing relationships**

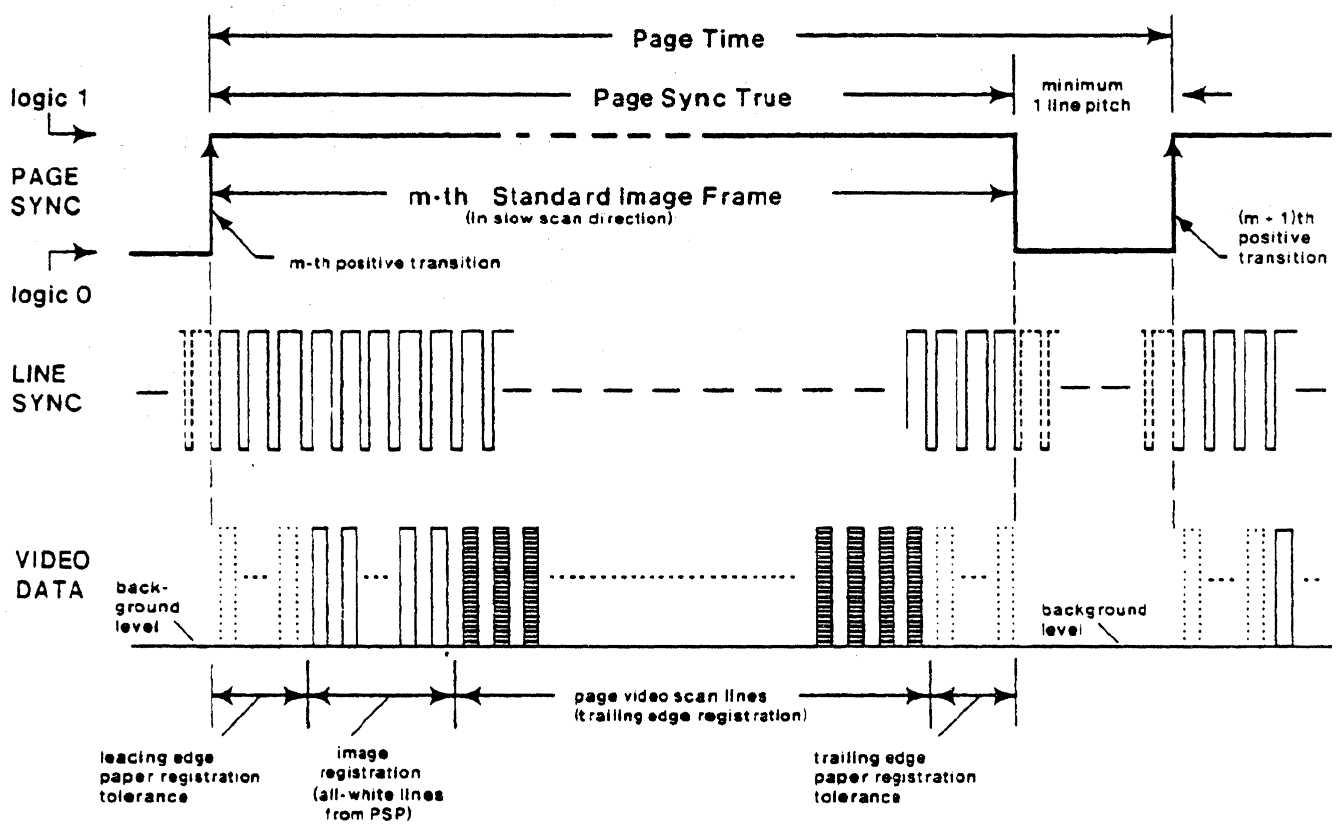


Figure 4-2. **Slow scan timing relationships for gapless printing IOT**

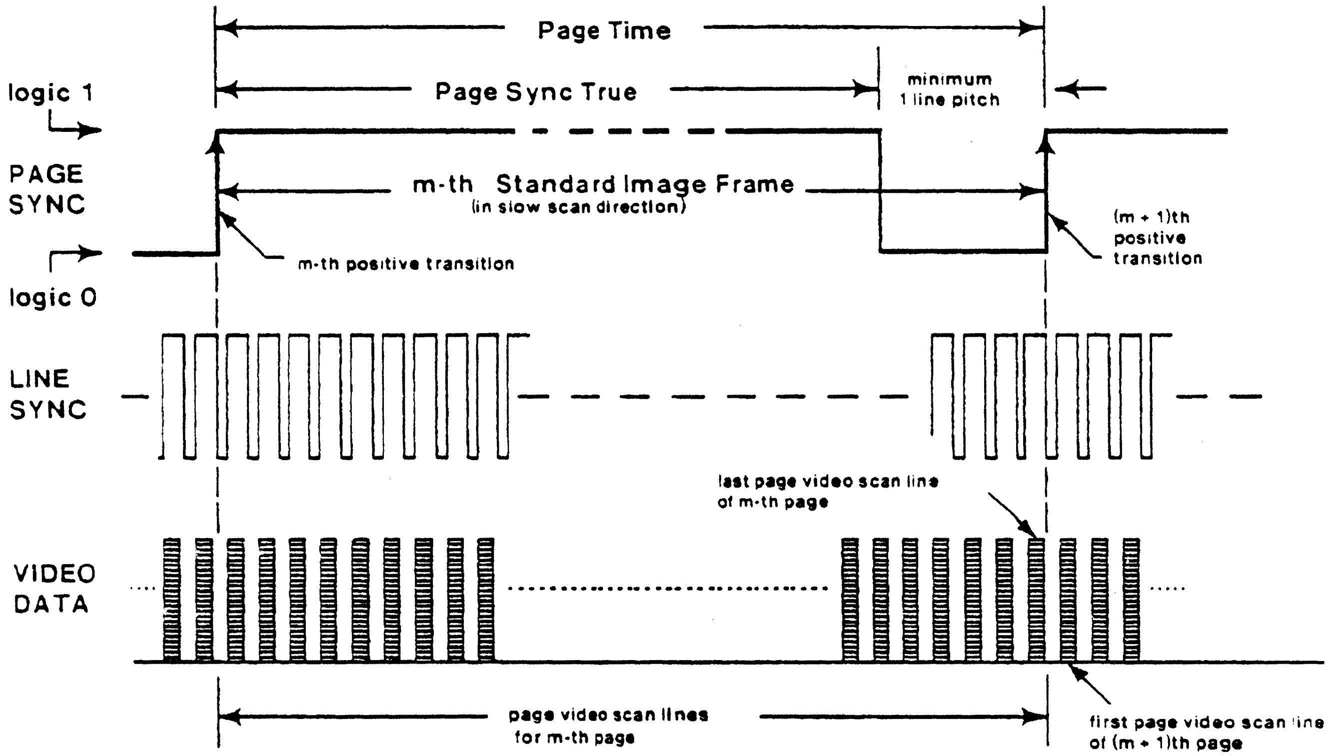


Figure 4-3. **Fast scan timing relationships, parallel mode**

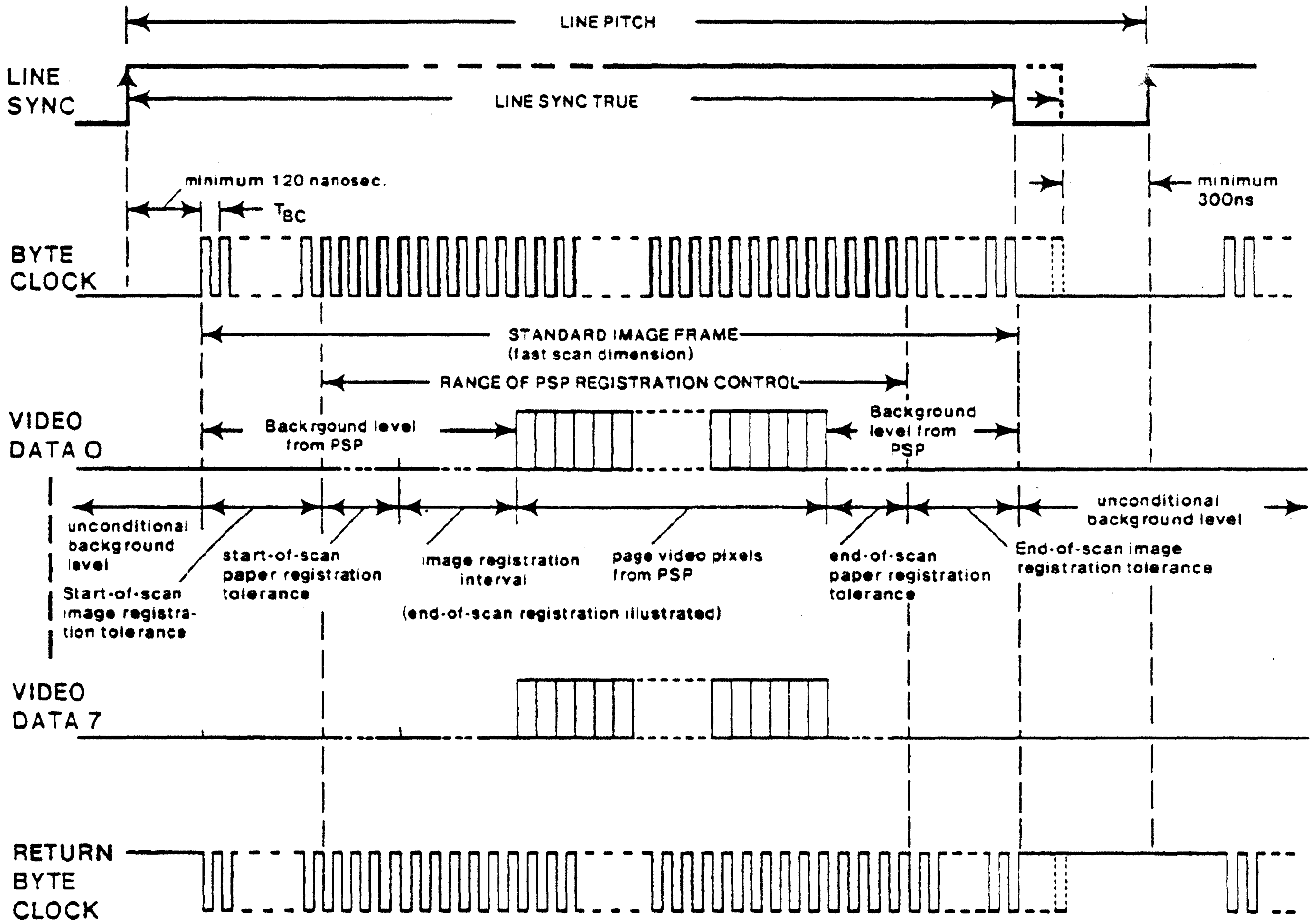
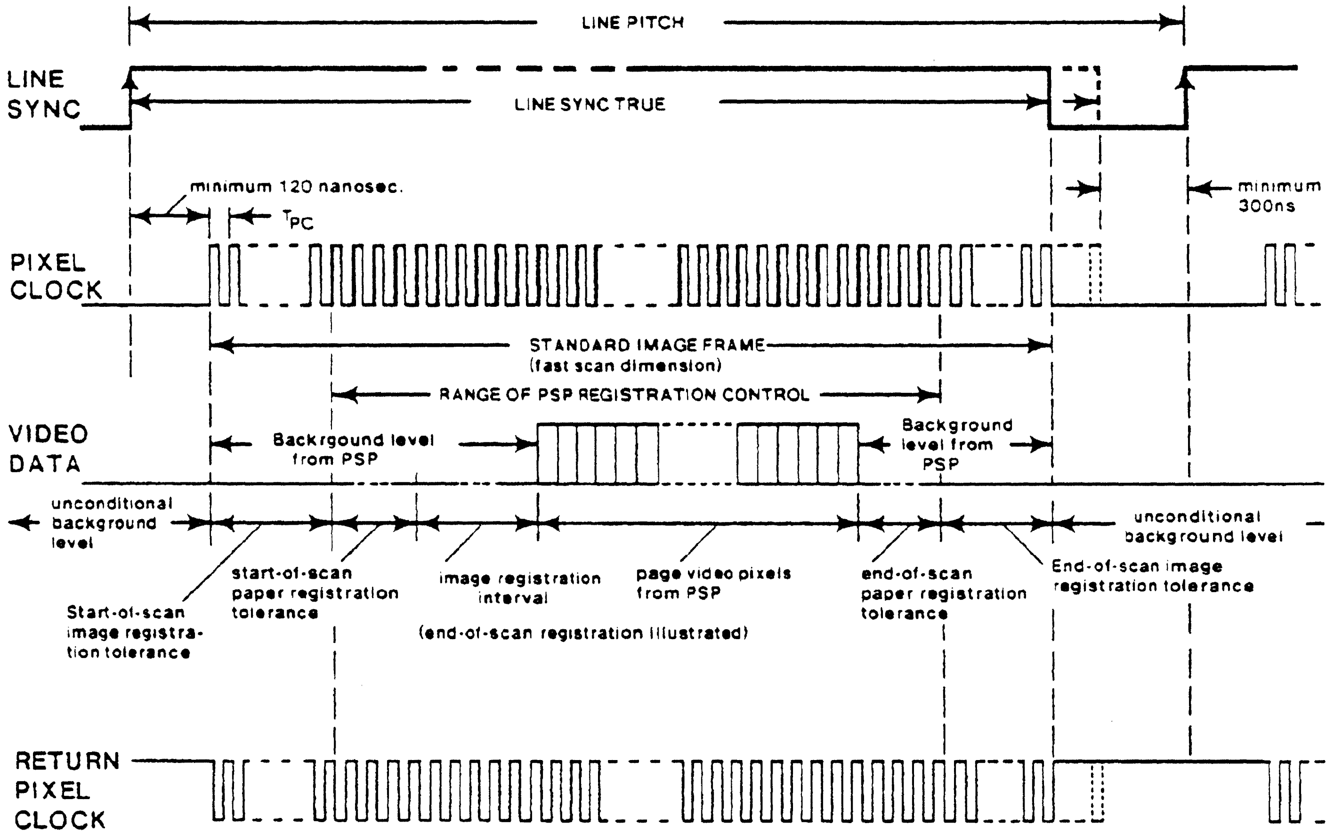


Figure 4-4. **Fast scan timing relationships, serial mode**



XEROX Private Data

5. Command/status communications

5.1 Introduction

This standard specifies hardware and software configurations, formats, and protocols, which constitute definitive parts of the *physical layer*, the *data link layer*, and the *client layer* of a layered architecture (refer to figure 5-1) for communicating Command and Status messages between PSPs and IOTs. The *physical layer* is responsible for the actual transmission and reception of signals in either direction, and delivery of the assembled messages to the associated processors. It also detects and reports certain exception conditions which can occur during transmission. The *data link* layer manages the message traffic and many of the exception conditions. The *client layer* constitutes the substantive logical aspects of the communication.

The interface between the client layer and the data link layer is via software. The interface between the data link layer and the physical layer is both a software and a hardware interface. It may be partially within a commercially available protocol controller. Such components can provide parts of both the data link layer and physical layer functions.

The command channel is also the means for down-loading blocks of non-image information from the PSP to the IOT. The status channel is the means for up-loading blocks of non-image information from the IOT to the PSP.

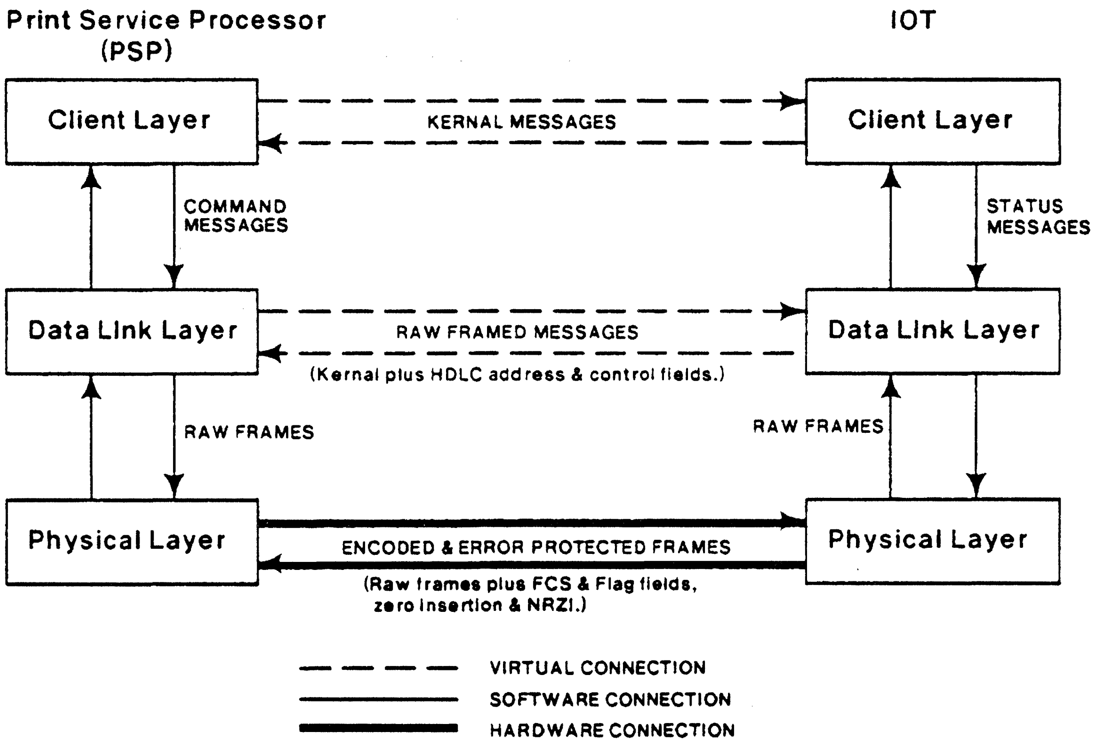
5.2 Physical layer

The following is a functional description of the physical layer implemented with a controller (refer to section 5.5) and interconnect hardware, including cables, cable drivers, and cable receivers. The nomenclature convention for the signals on the Command and Status channels is the same as for the Imaging Interface signals (refer to chapter 4).

5.2.1 Topology

The Command and Status communication interfaces shall constitute a duplex data link operating in the serial, synchronous transmission mode. Physically, the Command and Status data shall be carried (in opposite directions) on separate, serial, simplex, balanced interface circuits. Each transmitting function must have its own stable frequency source. Each receiving function must recover bit timing from the received data stream in order to reclock the received data. The channel characteristics and physical interfaces shall be identical for the Command function and the Status function, except for the direction of transmission and the location of the transmitting and receiving circuitry. The topology must be point-to-point, at the physical layer, i.e., only one PSP and one IOT interconnected. This restriction derives from the particular data link protocol adopted for this interface (refer to section 5.3).

Figure 5-1. **Layered architecture, Command/Status communications**



5.2.2 Bit stream encoding, ZBI, NRZI

The data within a frame, i.e., between flags (refer to section 5.3.3-a), shall undergo zero-bit-insertion (ZBI). In this encoding procedure, a binary 0 must be inserted by the transmitter after any succession of five contiguous 1s within a frame. ZBI prevents the occurrence of the flag pattern (01111110) anywhere within a frame. After the receiver recognizes a beginning flag, it removes any 0 which follows five contiguous 1s. Inserted and removed 0s are not included in the frame check sequence computation (a 1 that follows five 1s is not removed).

5.2.3 Bit rate[1]

The bit rate shall be 9.6, 19.2, or 57.6 Kilobits per second, ±0.01 percent. Command and Status channels must both use the same data rate, i.e., a transmitting function must always use the same data rate as its co-located receiving function. PSPs must implement all three data rates, and they must be programmable. IOTs may implement any or all of these data rates, and they may be programmable. The PSP is responsible for determining the data rate of the IOT from the IOT's transmissions, and must program its data rate to match.

5.3 Data link layer

5.3.1 Introduction

The data link layer consists of the PSP (or IOT) operating system's communications, timer, and interrupt handling code, in conjunction with data link functions performed by a controller. Together these facilities implement a subset of the HDLC (High-level Data Link Control) protocol, which is described in the following international standards:

ISO 3309, Data Communication—High-level Data Link Control Procedures—Frame Structure. [Ref. 1]

ISO 4335, Data Communication—High-level Data Link Control Procedures—Elements of Procedures. [Ref. 2]

ISO 6159, Data Communication—HDLC Unbalanced Class of Procedures. [Ref. 3].

These standards are to be considered a part of this generic specification, except as modified or restricted herein. The particular implementation prescribed in this generic specification utilizes the Unbalanced Class of procedures with the Asynchronous Response Mode and, with the selected optional functions, is designated [Ref. 3] Class UAC, 1, 2, 4, 5. The numbered options are identified later. Under this implementation, the PSP has the primary responsibility for controlling the data link. However, it provides for unsolicited frame transmission from both IOT and PSP in a two-way simultaneous (duplex) mode. The following is a discription of pertinent aspects of HDLC as it is to be applied at this Interface.

1 The term *baud* rate is equivalent since the transmission format is binary.

5.3.2 Station types

Two types of stations are defined for the unbalanced classes of procedures: a primary station, which sends orders and receives responses, and is responsible for link level error recovery; and secondary stations, which receive orders[2], send responses, and may initiate link level error recovery. The PSP is designated the primary station, and the IOT is designated the secondary station.

5.3.3 Frame structure

All transmissions are organized into frames. The frame format enables the receiver to determine where transmission starts and stops, the identity of the IOT receiving an order or sending a response, what actions are to be performed with the transmission, specific information for the receiving station, and data that is used to check that the frame was received without error. Each frame conforms to the format shown in figure 5.2. A frame consists of six fields in the following order: Beginning Flag sequence, Address(A), Control(C), Information(I), Frame Check Sequence(FCS), Ending Flag sequence. The Beginning and Ending Flag sequences are identical single-byte fixed patterns; the A and C fields are also one byte. The FCS field is two bytes. The Information field must consist of multiple bytes, or zero bytes. The latter case, i.e., Information field absent, forms a special case in which the Control field carries supervisory sequences to be interpreted at the Data Link level.

a) *Flag sequence:*

All frames shall start and end with the Flag sequence. Both receiving stations shall continuously hunt for this sequence. Thus, the flag is used for frame synchronization. The beginning flag serves as a reference for the position of the A and C fields, and initiates error checking. The ending flag terminates error checking. Both beginning and ending flags have the binary configuration 01111110. The ending flag for one frame may serve as the beginning flag for the next frame. Also, there may be multiple flags repeated between frames to maintain the active state. Zero bit insertion (refer to section 5.2.2) prevents the flag pattern from occurring anywhere else in the frame.

b) *Address field:*

In *order* frames, the address shall identify the IOT for which the order is intended. In *response* frames, the address shall identify the IOT from which the response originated. The default IOT address is 0000 0001 (binary). Any other address assigned to an IOT must be known to the PSP a priori.

c) *Control field:*

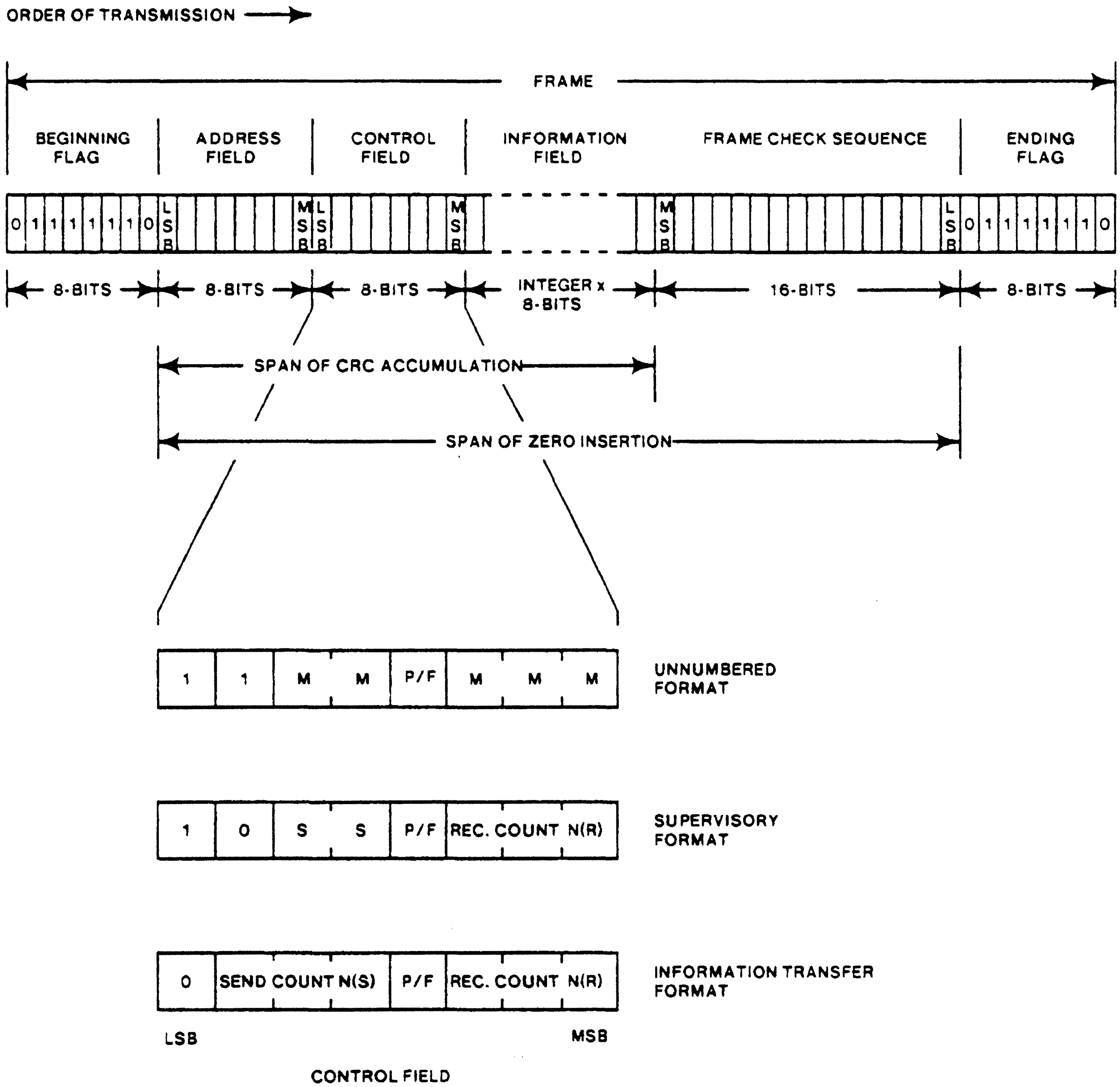
The Control field may contain an order or a reponse, and/or frame sequence numbers, to be processed at the data link level. The Control Field can be in one of three formats (refer to figure 5-2): *unnumbered, supervisory*, or *information transfer* format. The Control Field formats are further defined in sections 5.3.5, 5.3.6, and 5.3.8.

[2] In the references, these are called *commands*. In this generic specification, the term *command* is reserved for Client Layer messages originated by the PSP (refer to chapter 6).

Unnumbered format:

Unnumbered format frames are used for such functions as: initializing an IOT; controlling the response mode of an IOT; reporting certain procedural errors; and transferring data when the data is not to be checked as to its location in a sequence of frames.

Figure 5-2. **Command/Status frame structure**



Supervisory format:

Frames with Control Field in the supervisory format are used to acknowledge reception of (or to reject) preceding frames carrying information. They may also be used to convey ready or busy conditions, and to report frame numbering errors (a numbered frame received out of proper sequence).

Information transfer format:

Frames with Control Field in the information transfer format contain an information field which carries the PSP Command messages or the IOT Status messages (or carries blocks of data to or from the IOT). This control field, in addition to indicating the frame format, contains send and receive counts (Ns and Nr, respectively), which are used to insure that these frames are received in their proper order (Ns) and to confirm accepted information frames (Nr).

d) *Information field:*

The Information Field may carry a Client Layer Command message from the PSP, a Client Layer Status message from the IOT, or a block of data to or from the IOT. These messages or data blocks may be any sequence of bits whose length is an integral number of eight-bit bytes, not to exceed 124 bytes, i.e., the total Data Link frame, excluding flags, must not exceed 128 bytes. The actual Command/Status messages which are transmitted in the I-fields are defined in chapter 6.

e) *Transparency:*

The contents of the frame between the two flag sequences, consisting of the Address, Control, Information, and FCS Fields, shall be transparent to the flag detectors by virtue of the zero-bit-insertion coding described in section 5.2.2.

f) *Frame check sequence field:*

This field shall contain a 16-bit cyclic redundancy check (CRC) sequence that is the result of a computation on the contents of the A, C, and I fields at the transmitter. The receiver shall perform a similar computation and check the results against the known expected result. If an error is found, the frame shall not be accepted. (Refer to Ref. 1, section 3.6, and Annex A, for a full description of the CRC and discussion of implementation.)

g) *Order of bit and byte transmission:*

Addresses, orders, responses, and sequence numbers shall be transmitted least significant bit (lsb) first (for example, the first bit of the sequence number that is transmitted shall have the weight 2^0). The FCS shall be transmitted to the line commencing with the coefficient of the highest term.

The Most Significant Byte (MSB) of a Command or Status message is transmitted first in the Information Field. The least significant bit (lsb) of a Command or Status byte is transmitted first in the Information Field. For blocks of data, the first byte is taken from, or placed in, memory at the start address, and so on.

h) Inter-frame time fill:

Inter-frame time fill shall be accomplished by transmitting contiguous flags.

i) Invalid frame:

An invalid frame is defined as one that is not properly bounded by two flags, or one which is too short (for example, shorter than 32 bits between flags). Invalid frames shall be ignored. Thus, a frame which ends with an all "1" bit sequence equal to or greater than seven bits in duration shall be ignored.

5.3.4 IOT operational modes

Note: Under HDLC, operational modes are defined only with respect to the secondary station.

The IOT shall operate in either the Asynchronous Response Mode (ARM), the Asynchronous Disconnected Mode (ADM), or the Initialization Mode (IM).

ARM is a secondary station mode in which the IOT may initate transmission without receiving explicit permission from the PSP. The transmission may be used for information field transfer and/or for supervisory information (for example, to indicate the number of the next expected information frame, transition from a ready to a busy condition or vice versa, occurrence of an exception condition).

ADM is a secondary station mode in which the IOT is logically disconnected from the data link and is, therefore, not in operational status. (The IOT should, however, maintain the active channel state with the prescribed interframe time fill—refer to section 5.3.7). In this mode, the IOT has restricted asynchronous mode response capability. The IOT may initiate a response transmission at any time, but the transmission shall be limited to a request for logical connection to the PSP (DM), or a request for initialization (RIM) if the IOT determines that it is unable to function. (Refer to section 5.3.8 for definition of orders/responses.) In ADM, the IOT, if capable, shall act only on mode setting orders and XID. Other valid orders shall cause the IOT in ADM to send a disconnect mode response (DM), or, if unable to function, a request for initialization (RIM).

IM is an IOT mode invoked by a set initialization mode (SIM) order initiated by the PSP or sent by the PSP as the result of a request initialization mode (RIM) from the IOT. The IOT shall enter IM upon sending a UA in response to SIM, and shall then initialize its data link layer according to its own internal procedures. IM is a transient mode; the IOT automatically progresses to ADM when initialization is complete. The IOT power-on sequence must also cause the IOT to enter IM. No communication is allowed between the IOT and the PSP in IM. Note that IM as defined herein is not per HDLC.

An IOT data link state diagram is shown in figure 5-5.

5.3.5 Control Field formats

There are three Control Field formats, defined as follows, to perform numbered information transfer, numbered supervisory functions, and unnumbered control functions and information transfer. The generic formats are shown in the expanded portion of figure 5-2.

a) *Information transfer format—I:*

The I format is used to perform an information transfer. (Information may also be transferred in the unnumbered format, refer to (c), this section.) Each I frame has an N(S) sequence number and an N(R) sequence number.

b) *Supervisory format—S:*

The S format is used to perform link supervisory control functions such as acknowledge I frames, request retransmission of I frames, and to request a temporary suspension of transmission of I frames.

c) *Unnumbered format—U:*

The U format is used to send unnumbered information (UI) frames and to provide additional link control functions. This format contains no sequence numbers and, consequently, 5 "modifier" bit positions are available which allow definition of up to 32 additional order functions and 32 additional response functions. A total of 11 modifier codes are used to define 3 orders, 5 responses, and 3 with dual usage as orders or responses (refer to section 5.3.8).

5.3.6 Control Field parameters

a) *Modulus:*

The modulus shall be 8. Each I frame is sequentially numbered and may have the value 0 through 7. The sequence numbers cycle, modulo 8, through the entire range.

The maximum number of sequentially numbered I frames that the PSP or IOT may have outstanding (i.e., unacknowledged) at any given time is one (1). This restriction greatly simplifies the maintenance of information integrity and error recovery. It also means that a transmit buffer need hold only one frame at a time awaiting acknowledgement.

b) *Frame variables and sequence numbers:*

The PSP and IOT each maintain an independent send sequence number N(S) and receive sequence number N(R) on the I frames that they send and receive. Thus, the PSP maintains independent N(S) and N(R) counts for I frames sent to and received from the IOT. And, likewise, the IOT maintains independent N(S) and N(R) counts for I frames sent to and received from the PSP. This HDLC sequence numbering facilitates an unambiguous retransmission strategy and precludes any confusion during 2-way simultaneous frame transmission.

c) *Send state variable V(S):*

The send state variable denotes the sequence number of the next in sequence I frame to be transmitted. The send state variable can take on the value 0 through 7. The value of the send state variable is incremented by one with each successive I frame transmission, but cannot exceed N(R) of the last received frame by an increment (calculated modulo 8) of more than 1, because the number of outstanding frames is limited to one.

d) Send sequence number N(S):

Only I frames contain N(S), the send sequence number of transmitted frames. Prior to transmission of an in-sequence I frame, the value of N(S) is updated to equal the value of V(S). Note that with this relationship between V(S) and N(S), and the limit of one outstanding frame, N(S) of a transmitted frame cannot exceed N(R) of the last received frame.

e) Receive state variable V(R):

The receive state variable denotes the sequence number of the next in-sequence I frame to be received. The receive state variable can take on the values 0 through 7. The value of the receive state variable is incremented by the receipt of an error-free, in-sequence I frame whose send sequence number N(S) equals the receive state variable.

f) Receive sequence number N(R):

All I frames and S frames contain N(R), the expected sequence number of the next received I frame. Prior to transmission of an I or S frame, the value of N(R) is updated to equal the current value of V(R). N(R) indicates that the station transmitting the N(R) has correctly received the I frame numbered N(R)-1.

Note: Neither N(S) nor N(R) appears in the control field of the unnumbered format.

g) Poll/Final (P/F) bit:

All Control Field formats contain a P/F bit. In frames sent from the PSP, the P/F bit is referred to as the P bit. In frames sent from the IOT, the P/F bit is referred to as the F bit. Under HDLC/ARM, this bit plays an important role in maintaining the integrity of information exchange, and in error recovery. However, this function is not utilized in the PSP/IOT data link, because the restriction to one outstanding frame, and the direct physical transmission paths, render this sophistication unnecessary. Accordingly, the P/F bit shall be set to "0" in all frames at all times.

5.3.7 Data link channel states

a) Active channel state:

A channel is in an *active* condition when the PSP or the IOT is actively transmitting a frame, a single *abort* sequence, or interframe time fill. In the *active* state, the right to continue transmission is reserved.

Abort; aborting a frame is accomplished by transmitting at least seven contiguous "1" bits (without zero insertion, section 5.2.2). Receipt of seven contiguous "1" bits is to be interpreted as an abort, and the receiving station shall ignore the frame. (Note that a sequence of 15 or more "1" bits will be interpreted as an idle condition—refer to (b).

Interframe time fill; Interframe time fill is accomplished by transmitting continuous flags between frames. There is no provision for time fill within a frame.

b) Idle channel state:

Under HDLC, a channel is defined to be in an *idle* condition when a continuous transmission of at least 15 "1" bits is detected. The idle state is not to be intentionally invoked by the PSP or the IOT. If the idle state sequence is received, it is to be interpreted as a fault condition, which must be resolved at a higher level.

c) Transmission mode:

The physical layer provides a duplex transmission facility (section 5.2.1), and the data link must support two-way-simultaneous transmission of frames. Actual transmissions may occur as two-way-simultaneous (duplex) or two-way-alternate (half-duplex) frame sequences.

5.3.8 Order/response definitions

a) Summary:

The subset of HDLC *order* and *response* frames to be used for the interface data link is tabulated below, and the Control Field bit formats are shown in figure 5-3. Note that some are orders only, some are responses only, and some serve as both orders and responses.

Name	Acronym	Order	Response
I (Information) format			
Information	I	x	x
S (Supervisory) format			
Receive Ready	RR	x	x
Receive Not Ready	RNR	x	x
Reject	REJ	x	x
U (Unnumbered) format			
Unnumbered Information	UI	x	x
Request Initialization Mode	RIM		x
Set Initialization Mode	SIM	x	
Set Asynchronous Response Mode	SARM	x	
Disconnect Mode	DM		x
Disconnect	DISC	x	
Unnumbered Acknowledge	UA		x
Frame Reject	FRMR		x
Request Disconnect	RD		x
Exchange Identification	XID	x	x
Test	TEST	x	x

Figure 5-3. **Control Field bit formats**

		ORDER OF TRANSMISSION →								PRESENTATION ORDER							
FORMAT	ACRONYM	b0	b1	b2	b3	b4	b5	b6	b7	b7	b6	b5	b4	b3	b2	b1	b0
I	I	0	N(S)			P/F	N(R)			N(R)			P/F	N(S)			0
S	GENERIC	1	0	S	S	P/F	N(R)			N(R)			P/F	S	S	0	1
	RR	1	0	0	0	P/F	N(R)			N(R)			P/F	0	0	0	1
	REJ	1	0	0	1	P/F	N(R)			N(R)			P/F	1	0	0	1
	RNR	1	0	1	0	P/F	N(R)			N(R)			P/F	0	1	0	1
U	GENERIC	1	1	M	M	P/F	M	M	M	M	M	M	P/F	M	M	1	1
	UI	1	1	0	0	P/F	0	0	0	0	0	0	P/F	0	0	1	1
	RIM	1	1	1	0	F	0	0	0	0	0	0	F	0	1	1	1
	SIM	1	1	1	0	P	0	0	0	0	0	0	P	0	1	1	1
	SARM	1	1	1	1	P	0	0	0	0	0	0	P	1	1	1	1
	DM	1	1	1	1	F	0	0	0	0	0	0	F	1	1	1	1
	DISC	1	1	0	0	P	0	1	0	0	1	0	P	0	0	1	1
	UA	1	1	0	0	F	1	1	0	0	1	1	F	0	0	1	1
	FRMR	1	1	1	0	F	0	0	1	1	0	0	F	0	1	1	1
	RD	1	1	0	0	F	0	1	0	0	1	0	F	0	0	1	1
	XID	1	1	1	1	P/F	1	0	1	1	0	1	P/F	1	1	1	1
	TEST	1	1	0	0	P/F	1	1	1	1	1	1	P/F	0	0	1	1

(b1 and b5 are the lower order bits of N(S) and N(R), respectively.)

b) I (Information) format:

The I, Information, order/response is used by both the PSP and the IOT to transfer across the interface data link sequentially numbered frames containing an information field. The information field may be of any length up to the maximum of 124 bytes. The variable content of the I frame Control Field consists of two sequence numbers, N(S) and N(R) (refer to section 5.3.6 for a discussion of the sequence numbers).

Each correctly received I-frame shall be acknowledged by the receiving station at the earliest opportunity. Acknowledgement shall be via the value of Nr (incremented from the previous response) in a return I-frame or in an RR frame. Incorrectly received I-frames shall be handled as described in section 5.3.9.

c) S (Supervisory) format:

Supervisory order/responses are used by both the PSP and the IOT to perform numbered supervisory functions, such as acknowledgement, temporary suspension of information transfer, or error recovery.

Frames with the S format shall not contain an information field and, therefore, do not increment the sequence numbers at either the transmitter or receiver.

The individual S frame formats are identified by the two S bits in the Control Field. The variable content of an S frame Control

Field consists of the receive sequence number, N(R). N(R) indicates the sequence number of the next expected I frame to be received at the time of transmission, and consequently indicates that the I frame numbered N(R)-1 has been correctly received (refer to section 5.3.6 for discussion of the sequence numbers).

RR, Receive Ready:

The RR frame is used by the PSP or the IOT to:

- Indicate it is ready to receive an I frame;
- Acknowledge previously received I frame numbered N(R)-1 at the earliest opportunity;
- Clear a busy condition that was initiated by the transmission of RNR.

REJ, Reject:

This is an HDLC optional function which constitutes option 2 in the class of procedures designation (section 5.3.1). The REJ frame is used by the PSP or the IOT to request retransmission of the I frame numbered N(R). Only one REJ exception condition can be established at any given time in each direction of transmission because of the limitation of one unacknowledged frame. Another REJ may not be effected until the first REJ exception condition has been cleared. The REJ exception condition is cleared (reset) upon receipt of an I frame with an N(S) equal to the N(R) of the original REJ frame.

When used, REJ must be sent at the earliest opportunity following receipt of an out-of-sequence I-frame.

RNR, Receive Not Ready:

The RNR frame is used by the PSP or the IOT to indicate a busy condition, i.e., temporary inability to accept additional incoming I frames. I frame numbered N(R)-1 is acknowledged. I frame N(R) is not acknowledged. The acceptance status of this frame will be indicated in the subsequent exchange.

Indication that the busy condition has cleared and I frames will now be accepted is communicated by the transmission of an RR, REJ, SARM, UA, or an I frame.

An IOT receiving an RNR frame when in the process of transmitting (i.e., two-way simultaneous) shall stop transmitting I frames at the earliest possible time.

d) U (Unnumbered) format:

Unnumbered order/responses are used to extend the number of link control functions. Three of the formats are used only for orders, five are used only for responses, and three are used for both. Frames transmitted with the U format do not increment the sequence counts at either the transmitting or receiving station.

The individual U frame formats are identified by the five M (modifier) bits in the Control Field. There are no variables in a U frame Control Field.

UI, Unnumbered Information:

This is an HDLC optional function which provides the ability to exchange information fields without impacting the I frame sequence counts. It is option 4 in the class of procedures designation (section 5.3.1). A UI frame does not require acknowledgement. The information field of a UI frame may be of any length up to the maximum of 124 bytes.

RIM, Request Initialization Mode:

This is an HDLC optional function which provides IOT ability to request initialization. Together with SIM, discussed in the following section, it constitutes option 5 in the class of procedures designation (section 5.3.1).

Once an IOT has requested initialization from the PSP, i.e., established a RIM condition, additional orders subsequently received (other than SIM or DISC or, if capable, XID) shall be monitored only to detect a respond opportunity to retransmit RIM. No additional transmission shall be accepted or actioned until the condition is reset by the receipt of SIM or DISC. No information field is permitted with the RIM response.

SIM, Set Initialization Mode:

This is an HDLC optional function which provides PSP ability to initialize the IOT. Together with RIM, discussed previously, it constitutes option 5 in the class of procedures designation (section 5.3.1).

The SIM order causes the IOT to enter the Initialization mode (section 5.3.4). No information field is permitted with the SIM order. Prior to acting on this order, the IOT must confirm acceptance of SIM by transmitting the UA response. The UA response takes precedence over any I or S format response pending, and must be made at the first opportunity. The IOT may ignore all frames received following this order until UA has been sent. Upon acceptance of this order, the IOT send and receive variables are set to zero.

Previously transmitted I frames that are unacknowledged when this order is actioned remain unacknowledged. Whether the station retransmits unacknowledged outstanding I frames or not must be decided at a higher level.

SARM, Set Asynchronous Response Mode:

The SARM order is used to place the IOT in the asynchronous response mode (ARM). No information field is permitted with the SARM order. The IOT confirms acceptance of SARM by transmission at the first respond opportunity of a UA response. The UA response takes precedence over any I or S format response pending, and must be made at the first opportunity. The IOT may ignore all frames received following this order until UA has been sent. Upon acceptance of this order, the IOT send and receive variables are set to zero.

Previously transmitted I frames that are unacknowledged when this order is actioned remain unacknowledged. Whether the station retransmits unacknowledged outstanding I frames or not must be decided at a higher level.

DM, Disconnect Mode:

The DM response is used by the IOT to report that it is logically disconnected from the link, and is by definition in ADM. An IOT in ADM may send a DM response at any respond opportunity. The DM response is sent by the IOT in ADM, to inform the PSP that it is still in ADM and cannot action a set mode order. No information field is permitted with the DM response.

An IOT in ADM shall monitor received orders to detect a respond opportunity in order to (re)transmit DM (or RIM, XID, or RD as appropriate), i.e., no orders (other than XID) are accepted until the disconnected mode is terminated by the receipt of SARM or SIM.

DISC, Disconnect:

The DISC order is used to inform the IOT that the PSP is suspending operation and that the IOT should enter ADM. No information field is permitted with the DISC order. Prior to acting on this order, the IOT must confirm acceptance of DISC by transmitting the UA response. The UA response takes precedence over any I or S format response pending, and must be made at the first opportunity. The IOT may ignore all frames received following this order until UA has been sent.

Previously transmitted I frames that are unacknowledged when this order is actioned remain unacknowledged. Whether the station retransmits unacknowledged outstanding I frames or not must be decided at a higher level.

UA, Unnumbered Acknowledge

The UA response is used by the IOT to acknowledge the receipt and acceptance of the U format orders defined above. Received U format orders are not actioned until the UA response is transmitted. No information field is permitted with the UA response.

FRMR, Frame Reject:

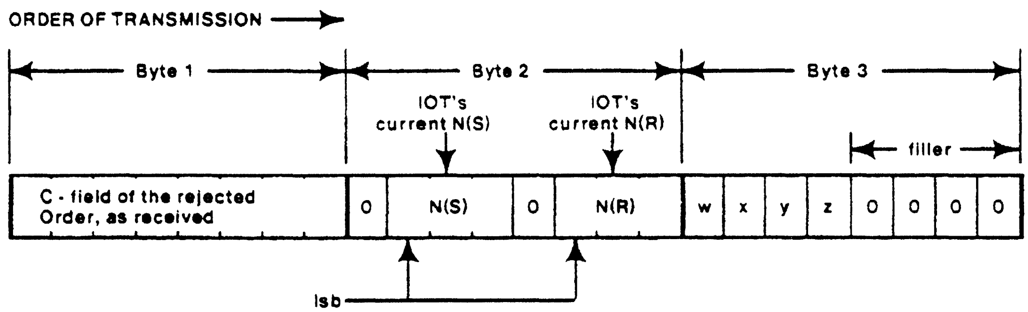
Note: The CMDR nomenclature used for this function in Ref. 2 is believed to be obsolete.

The FRMR response is used by an IOT in ARM to report to the PSP that certain exception conditions have occurred upon receipt of a frame without FCS error. The IOT must respond at the first opportunity. A three-byte information field which identifies the exception condition(s) is returned with this response. Included in the information field are the IOT's current N(S) and N(R), and the Control Field of the rejected order (as received). The format of the information field is shown in figure 5-4. The exception conditions and the method of reporting them are as follows:

- Bit "w" is set to "1" to indicate that the Control Field received and returned in the first byte of the information field is invalid or not implemented.
- Bit "x" is set to "1" to indicate that the Control Field received and returned in the first byte of the information field is considered invalid because the frame contained an information field which is not permitted with the received Control Field format. Bit "w" must be set to "1" if bit "x" is set to "1".
- Bit "y" is set to "1" to indicate that the information field received exceeded the maximum length allowed (124 bytes) or the capacity of the IOT's receiving buffer. This applies to UI and XID frames as well as to I frames.
- Bit "z" is set to "1" to indicate that the Control Field received and returned in bits 1 through 8 (first byte of the information field) contained an invalid N(R).

An invalid N(R) is defined as one which points to an I frame which has previously been transmitted and acknowledged, or to an I frame which has not been transmitted and is not the next sequential I frame pending transmission.

Figure 5-4. **Information field format of the FRMR response, as transmitted**



The w, x, y, and z bits in the information field of FRMR may all be set equal to zero, indicating an unspecified rejection of the order for one or more of the conditions cited above.

RD, Request Disconnect

This is an HDLC optional function which provides the IOT ability to request logical disconnection, i.e., transition to ADM. It is option 1B. Together with XID, discussed in the following section, it constitutes option 1 in the class of procedures designation (section 5.3.1). No information field is permitted with the RD response.

An IOT which has sent an RD response and receives an order frame (or frames), other than DISC shall accept the order frame (or frames) if it is able to do so. If the IOT accepts the non-DISC order frame (or frames), it shall follow the normal procedures to respond to the PSP. IOT acceptance of a frame other than DISC after sending an RD response shall cancel the RD response.

If the IOT still wishes to be placed in ADM, it shall re-issue the RD response. If the IOT cannot accept the non-DISC frames due to internal problems, it may respond with RD again.

XID, Exchange Identification:

This is HDLC option 1A. Together with RD, it constitutes option 1 in the class of procedures designation (section 5.3.1).

The XID order is used by the PSP to solicit identification of the IOT. An information field may be included in the order frame to convey identification and certain characteristics of the PSP.

The XID response is required from the IOT. An information field is included in the response frame to convey identification and certain characteristics of the IOT. The IOT must respond in any mode, unless a UA response to a mode setting order is pending, or a FRMR condition exists.

TEST, Test:

The TEST order is used by the PSP to solicit a TEST response from the IOT. The IOT must respond in any mode. If an information field is included in the order, it is returned in the response. If the IOT has insufficient buffering available for the information field, a TEST response with no information field is returned.

Note: HDLC does not prescribe a TEST command. This command is taken from SDLC, Ref. 4.

5.3.9 Exception reporting and recovery

This paragraph describes the error recovery procedures which are available to effect recovery following the detection/occurrence of an exception condition at the data link level. Exception conditions described are those situations which may occur as the result of transmission errors, station malfunction, or operational situations. Note that until the sending station receives acknowledgement of a transmitted frame, it must retain the original frame in memory, in case the need for retransmission should arise.

a) Busy:

The busy condition results when a station is temporarily unable to receive or continue to receive I frames due to internal constraints, for example, buffering limitations. In this case, an RNR frame is transmitted from the busy station. Traffic pending transmission may be transmitted from the busy station prior to or following the RNR. Indication that the busy condition has cleared and I frames will now be accepted is communicated by the transmission of an RR, REJ, SARM, UA, or an I frame.

b) N(S) Sequence error:

An N(S) sequence exception condition occurs in the receiver when an I frame received error free (no FCS error) contains an N(S) that is not equal to the receive state variable V(R) at the receiver. The receiver does not acknowledge (increment its receive state variable) the frame causing the sequence error, or any I frames which may follow, until an I frame with the correct N(S) is received. The information field of all I frames received whose N(S) does not equal the receive state variable V(R) will be discarded.

A PSP or IOT which receives an I frame having a sequence error, but FCS-error free, shall accept the control information contained in the N(R) field in order to receive acknowledgement of a previously transmitted I frame. Therefore, the retransmitted I frame may contain an N(R) field that is updated and, therefore, different from that contained in the originally transmitted I frame.

The station receiving an I-frame with N(S) not equal to its V(R) shall respond with an I-frame, or REJ, with N(R) still equal to its (unincremented) V(R). The sending station must then transmit (or retransmit) the frame with the required N(S), i.e., matching the N(R) which caused the exception condition. If there has been a procedural error at either end so that N(S) remains unequal to N(R) in the retransmission and the exception condition is not cleared, the problem must be resolved at a higher level. Resolution may involve sending SARM to reset the IOT's sequence variables to zero.

c) FCS error:

Any frame received with an FCS error is not accepted by the receiver. The frame is discarded, and no action is taken as the result of that frame. As a result, the sending station will experience an acknowledgement time-out and will commence retransmission as described in e).

d) Order Rejection (FRMR):

An order rejection condition is established by the IOT upon the receipt of an error-free order frame which contains an invalid Control Field, an invalid frame format, an invalid N(R), or an information field which exceeds the maximum length allowed (124 bytes), or exceeds the capacity of the IOT's receiving buffer.

At the PSP, this exception condition must be resolved at a higher level. At the IOT, this exception condition is reported via a FRMR response for appropriate action by the PSP. Once an IOT has established a FRMR exception, no additional I frames are accepted, except for examination of the N(R) field, until the condition is reset by the PSP. The FRMR condition may be reset by reception of a mode setting order or a DISC order.

e) *Time-out and retransmission strategy:*

In order to detect a no-reply or a lost-reply condition, the PSP and IOT shall each employ a time-out and retransmission strategy as follows:

i) The sending station of the PSP and the IOT data link layers shall each start an acknowledge time-out at the end of each outgoing frame which requires acknowledgement (via either receive-sequence-number or via UA). The duration of the acknowledge time-out shall be specific to the design of the IOT or PSP containing the receiving station and should be based on the frame processing time of the particular receiving station. It shall be as short as possible, but long enough to guarantee that correctly received frames are acknowledged, and, in any case, must be longer than 5 milliseconds. Note that the duration of the acknowledge time-out may be different for the two directions of transmission on a given duplex data link. The receiving stations' acknowledgement time requirements must be conveyed as the parameter DataLinkAckTime in the Client Layer messages, PspConfiguration, and IotConfiguration (refer to sections 6.3.2 and 6.3.3).

ii) Acknowledgement must be sent by the receiving station at the first opportunity, and the sending station's acknowledgement time-out must be stopped and reset upon reception of the acknowledgement.

iii) If an acknowledge time-out reaches completion, the sending station may retransmit the unacknowledged frame in a continuing attempt to obtain acknowledgement. The number of retransmissions and the interval over which the retransmissions occur shall be determined by the sending station, according to its own operational limitations, and may, for example, depend on the length of the frame and/or the particular machine state. The retransmission strategy of the sending station need not be known to the receiving station a priori.

iv) If a sending station terminates its retransmission strategy without success, the situation must be reported to a higher level for resolution.

Note: It is possible for an interface incompatibility to exist in regard to acknowledge time, e.g., the receiving function of a PSP might have a guaranteed acknowledgement time too long to support the real-time operation of an IOT with which it is proposed to interface. In that case, the acknowledgement time is to be considered part of the performance criteria which establishes the peer relationship in the compatibility requirement stated in the first paragraph of chapter 2, "Introduction."

5.4 Procedures (refer to figures 5-5, 5-6, and 5-7)

5.4.1 Initializing

With both units in the power-off state, the first action is to manually turn on the PSP AC power. The DC power is sequenced on automatically, and the power-on reset causes the PSP firmware and software to execute its initialization procedures. This will cause the IOT AC power to be turned on automatically via the **power control** interface signal. (Alternatively, the IOT power-on may be initiated manually at the PSP, or it may be accomplished manually at the IOT.) The power-on condition of the IOT will be verified to the PSP by the power monitor interface signal, if implemented (refer to section 3.2.5). The IOT DC power is sequenced on automatically, and the power-on reset causes the IOT firmware and software to execute its initialization procedures. The initialization procedures of the PSP and IOT also initialize the physical layers and the data link layers of their respective Command/Status interfaces. This means that the hardware and software is initialized as required to perform the signalling and protocol functions prescribed in sections 5.2 and 5.3. When initialization is complete, the IOT data link shall automatically advance to the asynchronous disconnect mode (ADM), and each direction of the interface channel shall be in the active state with each transmitter transmitting contiguous flags. And each receiver, as a result of first placing its bit timing recovery circuits in the search mode, should be locked on to the incoming signal and be looking for valid frames. Note that the IOT will transmit at its fixed or set data rate, and the PSP must detect this rate and adjust its send and receive data rates to match.

The PSP and IOT may exchange certain unnumbered frames while the IOT is in ADM, for example, RIM and/or SIM, and XID.

5.4.2 Establishing the data link

When ready to establish the data link, the PSP shall transmit SARM. The IOT, upon receiving SARM correctly, shall send UA at its first opportunity, set its send and receive variables to zero, and assume the asynchronous response mode (ARM). If the UA response is correctly received, the data link is established and the PSP shall set its send and receive variables to zero.

If the UA response is not received correctly by the PSP, or is not sent because SARM was not received correctly by the IOT, the PSP may enter the retransmission mode when its acknowledge time-out is completed. If the recovery procedure is not successful, the problem must be resolved at a higher level.

5.4.3 Exchange of information

Information frames, supervisory frames, and unnumbered frames are exchanged as prescribed in section 5.3, and illustrated in figure 5-7. Further examples of typical exchanges, with and without errors, are given in Ref. 2, Annex C, under C.4, "Examples of asynchronous response mode (ARM) 2-way simultaneous (FDX) transmission" (refer to chapter 6 for definition of Client Layer messages indicated informally in figure 5-7).

5.4.4 Logically disconnecting the data link

The PSP shall send DISC. The IOT, upon receiving DISC correctly, shall send UA at its first opportunity and then enter the asynchronous disconnect mode (ADM). The IOT transmits contiguous flags to maintain its half of the duplex channel in the active state. If the UA response is correctly received by the PSP, the PSP shall consider that the IOT is in ADM. The PSP transmits contiguous flags to maintain its half of the duplex channel in the active state.

If the UA response is not received correctly by the PSP, or is not sent because DISC was not received correctly by the IOT, the PSP may enter the retransmission mode when its acknowledge time-out is completed. The IOT may respond with either UA or DM during the recovery procedure. If the recovery procedure is not successful, the problem must be resolved at a higher level.

The PSP and IOT may exchange certain unnumbered frames while the IOT is in ADM, for example, RIM and/or SIM, and XID.

5.4.5 Shutting down the data link

The interface channel remains in the active state in both directions until DC power to the interface circuits is removed by the automatic power-down sequence, which may be initiated manually or automatically.

5.5 Implementation

This specification does not require a particular physical implementation of either the physical layer or the data link layer; however, there are a number of commercially-available multi-protocol serial controller (MPSC) chips which perform HDLC functions, as well as physical layer functions. Many of the functions and their parameters are programmable. Typically, these chips supply most or all of the following:

- One or two full duplex serial data channels with TTL interfaces.
- Byte-wide, bi-directional data bus interface.
- Vectored interrupt for controlling transmit and receive data transfer, and for reporting exception conditions.
- Automatic flag generation and detection.
- Automatic address recognition.
- Automatic CRC generation and checking.
- Frame assembly, byte-parallel to byte-serial, in transmission mode.
- Frame disassembly, byte-serial to byte-parallel, in receive mode.
- Zero-bit-insertion and removal.
- NRZI encoding and decoding.
- Programmable data rates.

- Bit-clock recovery from the received data stream, including sync search.
- Automatic determination of received data rate.

5.6 Client layer

The client layer is a software entity which interfaces with the data link layer and allows remote processes to communicate with each other without knowledge of the intervening communications protocols. Refer to chapter 6 for a full description of the Client Layer Protocol.

Figure 5-5. **IOT data link state diagram**

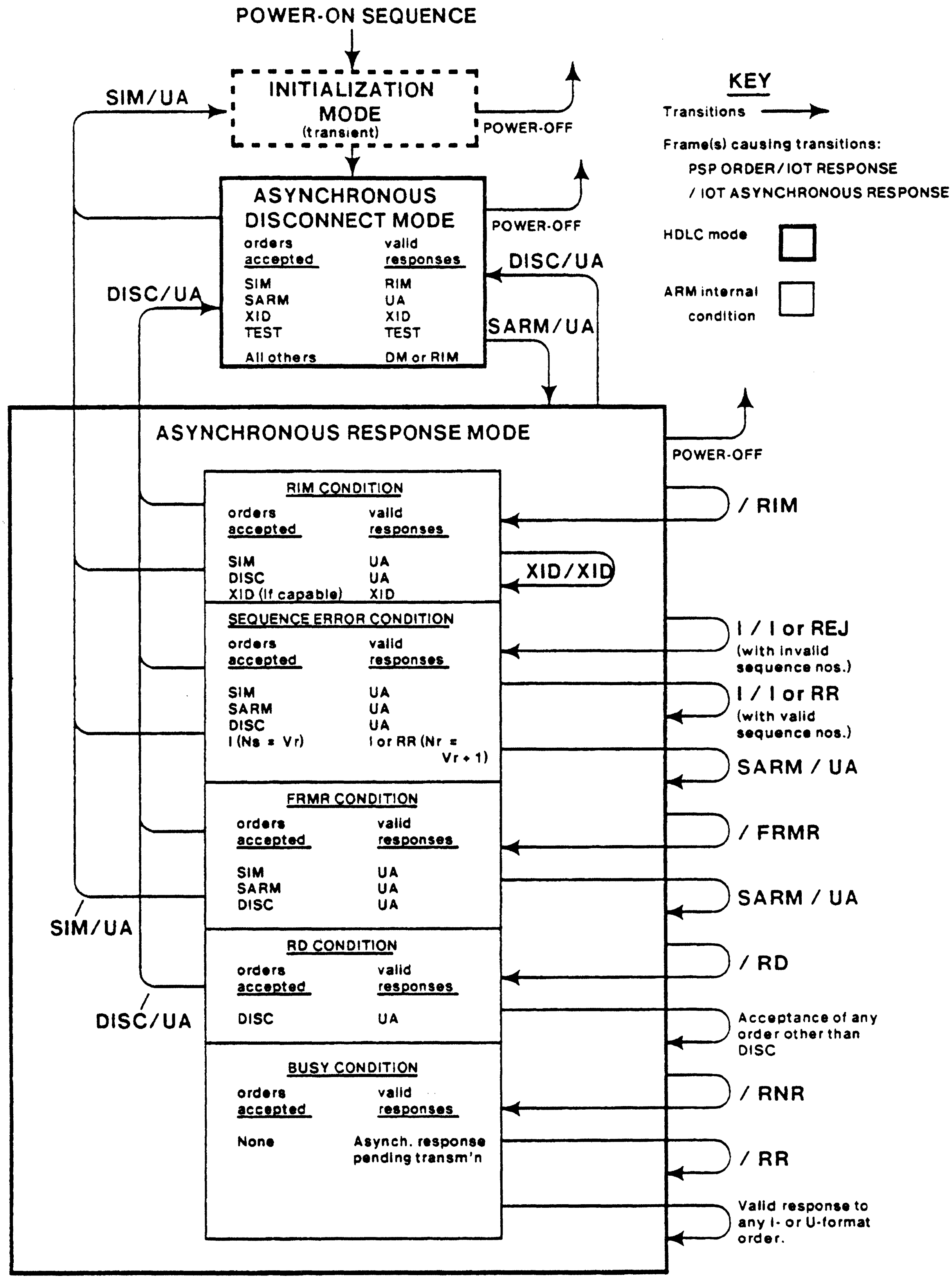


Figure 5-6. **Data link initialization**

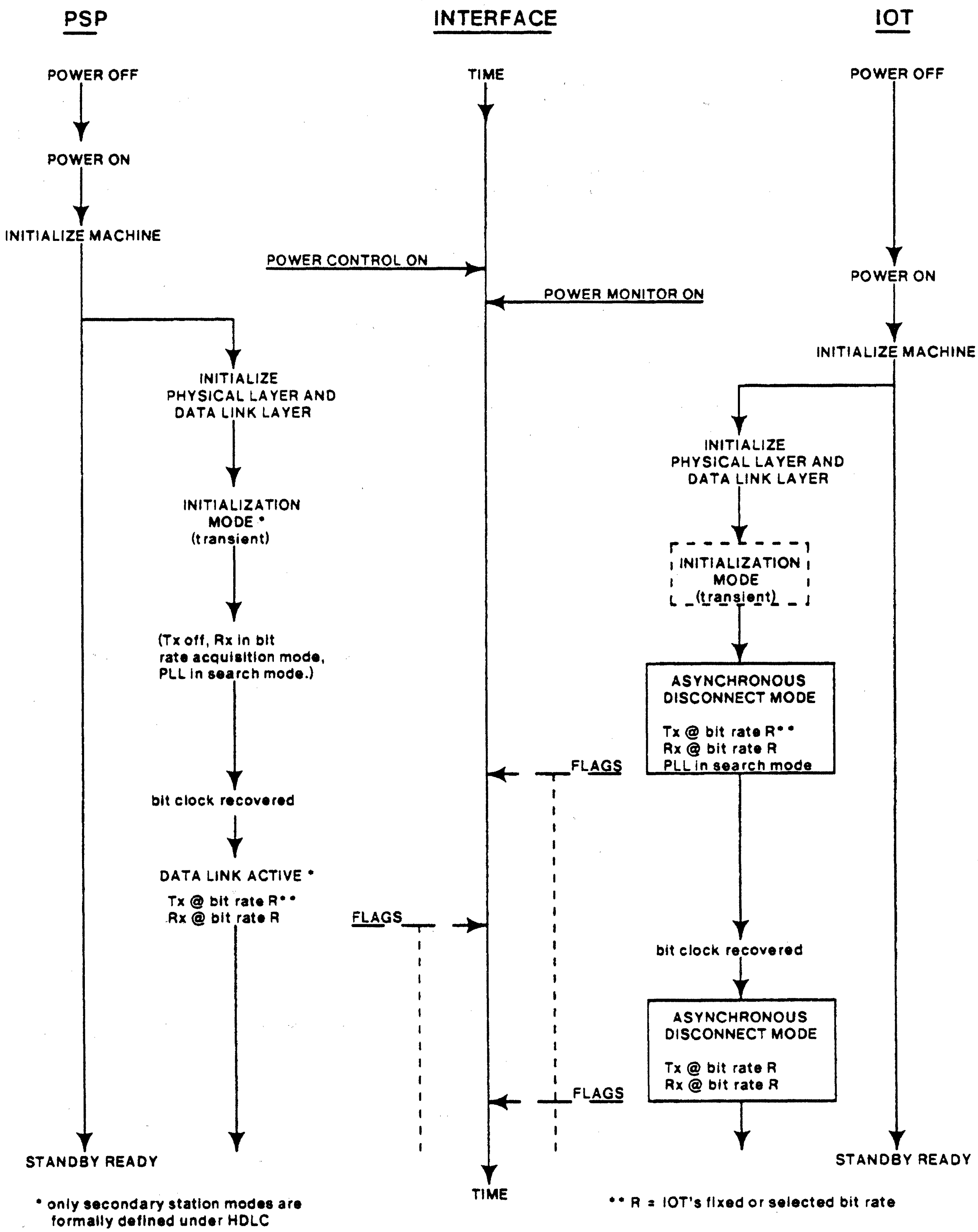
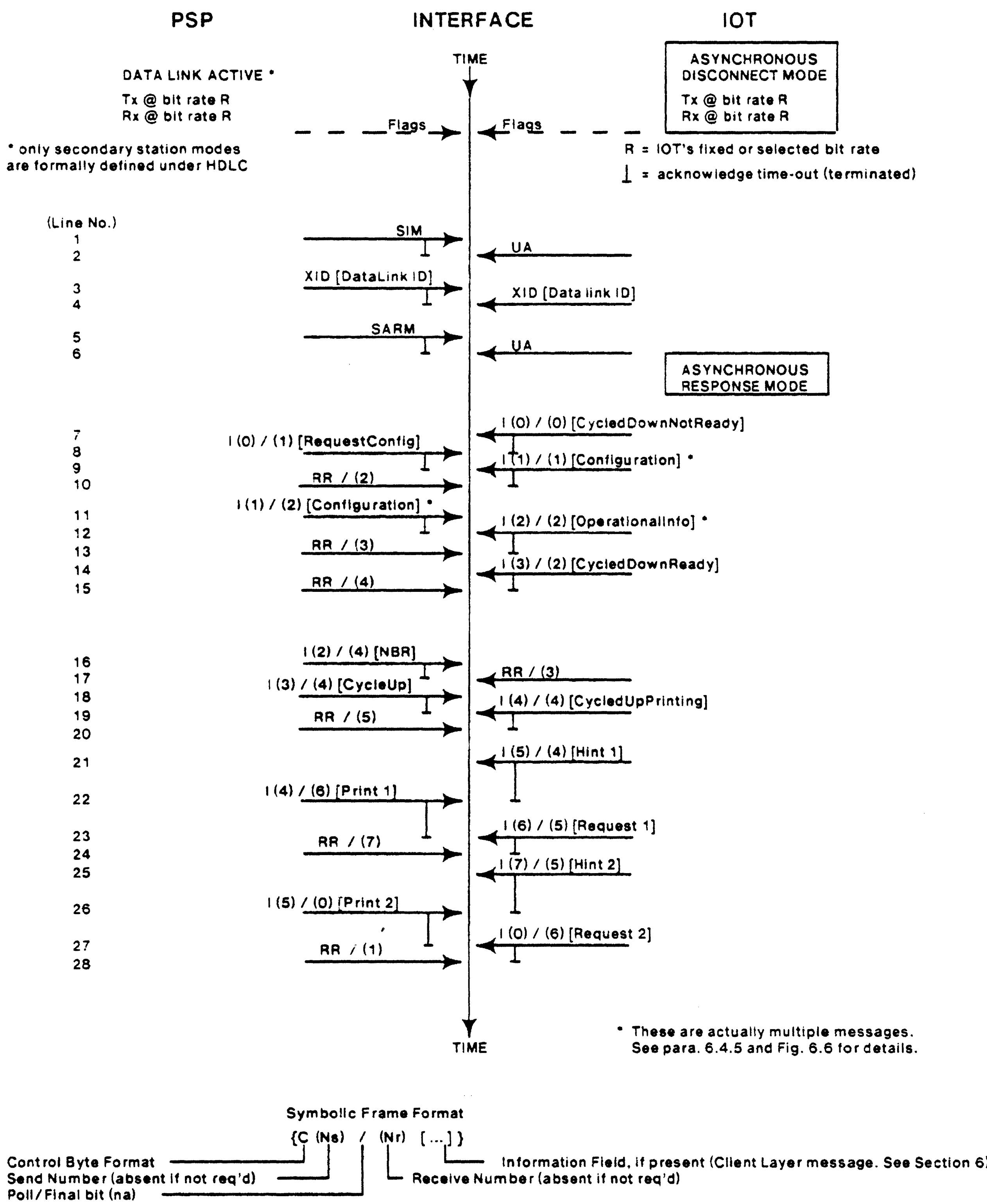


Figure 5-7. **Data link communication sequence**



XEROX Private Data

6. Client layer protocol

The Client layers of the PSP and IOT software will communicate via the set of Command and Status messages listed and defined below. These messages constitute the information field of the HDLC frame defined in chapter 5. In general, they should be sent in the numbered Information format. Possible exceptions are cited below in the application notes of the pertinent messages.

6.1 Identification code assignments, and page references

Code	Command	Page	Code	Status	Page
01	PSPCONFIGURATION	6–5	81	IOTCONFIGURATION	6–38
02	PSPMETARESET	6–7	82	IOTMETARESET	6–49
03	PSPNEXTBANKREQUEST	6–8	83	IOTVIDEOHINT	6–50
04	PSPPRINT	6–18	84	IOTVIDEOREQUEST	6–52
05	(u/a)		85	(u/a)	
06	PSPREADIOTMEMORY	6–20	86	IOTWRITEMEMORYTOPSP	6–53
07	PSPREADIOTSTATE	6–20	87	IOTSTATEINFO	6–54
08	PSPREADIOTOPERATIONALINFO	6–21	88	IOTOPERATIONALINFO	6–58
09	PSPREQUESTIOTDIAGNOSTIC	6–22	89	IOTDIAGNOSTICRESPONSE	6–63
0A	(u/a)		8A	(u/a)	
0B	PSPREJECTIOTMESSAGE	6–24	8B	IOTREJECTPSPCOMMAND	6–65
0C	PSPSHEETBANKABORT	6–27	8C	IOTSHEETDELIVERED	6–68
0D	PSPWRITEIOTMEMORY	6–29	8D	IOTREQUESTMEMORY	6–70
0E	PSPREQUESTIOTACTION	6–30	8E	IOTSWITCHINFO	6–71
0F	PSPREQUESTIOTSTATECHANGE	6–35	8F	(u/a)	

6.2 Summary of parameter assignments

6.2.1 PSP to IOT Command messages

COMMAND (ID hex)	Number of Parameters	Total no. Param. Bytes	Parameter names (no. of bytes)
PSPCONFIGURATION (01) Response: IOTCONFIGURATION or none	2	2	Command(1), Data(1)
PSPMETARESET (02) Response: none	1	1	PspMessage(1)
PSPNEXTBANKREQUEST (03) Response: none	21	27	PlateMode(1), SheetNumber(2), NumberOfCopies(2), NextBankTaskInfoA(1), NextBankTaskInfoB(1), NextBankTaskInfoC(1), BindexerFillLimit(1), SorterBindestination(1), StitchAPosition(1), StitchBPosition(1), FeederPermissions(1), DestinationPermissions(1), ManualFeederPermissions(1), ManualDestinationPermissions(1), PaperWidth(2), PaperLength(2), PaperType(1), FutureFinishingOptions(2), JobNumber(1). ContrastAction(1), ContrastData(2)
PSPPRINT (04) Response: IOTVIDEOREQUEST	4	6	PlateNumber(1), SheetNumber(2) CopyNumber(2), JobNumber(1)
PSPREADIOTMEMORY (06) Response: IOTWRITEMEMORYTOPSP	3	9	Spare(1), StartAddress(4), EndAddress(4)
PSPREADIOTSTATE (07) Response: IOTSTATEINFO	0	0	
PSPREADIOTOPERATIONALINFO (08) Response: IOTOPERATIONALINFO	1	1	InformationType(1)
PSPREQUESTIOTDIAGNOSTIC (09) Response: IOTDIAGNOSTICRESPONSE	4	7	Command(1), Utility(2), ParameterID(2), ParameterValue(2)
PSPREJECTIOTMESSAGE (0B) Response: IOTSTATEINFO, or IOTOPERATIONALINFO	8	10-13	Reason(1), IotStatusRejected(1), IotStatusParameterNumber(1), SheetNumber(2), CopyNumber(2), JobNumber(1), Spare(1) IotStatusParameterValue(1 - 4)
PSPSHEETBANKABORT (0C) Response: IOTSTATEINFO	4	6	AbortType(1), SheetNumber(2), CopyNumber(2), JobNumber(1)
PSPWRITEIOTMEMORY (0D) Response: none	4	7-123	EndOfBlock(1), StartAddress(4), Length(1), Data(1-117)
PSPREQUESTIOTACTION (0E) Response: IOTOPERATIONALINFO	4	5	Device(1), Set(1), Action(1), Data(2)
PSPREQUESTIOTSTATECHANGE (0F) Response: IOTSTATEINFO	1	1	ChangeRequested(1)

6.2.2 IOT to PSP Status messages

STATUS (ID hex)	Number of Parameters	Total no. Param. Bytes	Parameter names (no. of bytes)
IOTCONFIGURATION (81)	2	5,10,11, or 17	InformationType(1), DataField(4,9,10, or 16)
IOTMETARESET (82)	2	3	IotStatus(2) ResetMode(1)
IOTVIDEOHINT (83)	4	6	PlateNumber(1), SheetNumber(2), CopyNumber(2), JobNumber(1)
IOTVIDEOREQUEST (84)	4	6	PlateNumber(1), SheetNumber(2), CopyNumber(2), JobNumber(1)
IOTWRITEMEMORYTOPSP (86)	4	7-123	EndOfBlock(1), StartAddress(4), Length(1), Data(1-117)
IOTSTATEINFO (87)	2	2	MachineState(1), Substates(1)
IOTOPERATIONALINFO (88)	2	7 or 2-123	InformationType(1), DataField(6, or 1-122)
IOTDIAGNOSTICRESPONSE (89)	4	7	Status (1), Utility(2), ParameterID(2), ParameterValue(2)
IOTREJECTPSPCOMMAND (8B)	8	10-13	IotReason(1), PspCommandRejected(1), PspCommandParameterNumber(1), SheetNumber(2), CopyNumber (2), JobNumber(1), Spare(1), PspCommandParameterValue(1-4)
IOTSHEETDELIVERED (8C)	6	8	Integrity(1), SheetNumber(2), CopyNumber(2), Destination(1), SorterBin(1), JobNumber(1)
IOTREQUESTMEMORY (8D)	3	9	Spare(1), StartAddress(4), EndAddress(4)
IOTSWITCHINFO (8E)	2	2	Switch(1), Action(1)

6.3 Detailed description of Command and Status messages

6.3.1 Presentation format

The following sections define the high-level Command and Status messages passed between the PSP and the IOT client layers. The notation used is as follows:

Special characters:

::=	is defined to be
[a..b]	the interval a through b
|	or

-- a comment follows

Uppercase: denotes a keyword constant which is fixed in value.
(Example MACHINE ::= 1)

Lowercase: denotes a variable. (Example FaultState ::= [1..25])

The Command and Status message fields are composed of both byte-oriented and non-byte-oriented operands. When the operands are represented as numerical entities, normal English reading custom is followed, i.e., the left most digit is the most significant and the right most digit is the least significant, whether in hexadecimal or binary notation. The messages are packed so that non-byte-oriented operands fall on word boundaries in order to facilitate byte swapping, which is inevitable when communicating between processors with different internal byte ordering schemes.

Note that the bytes of the message fields are numbered in order of transmission, i.e., most significant byte first, which is the normal order of reading the entity when written as a number. The bit definitions are also numbered in order of transmission (within a byte), i.e., least significant bit first, which is opposite to the normal order of reading the entity when written as a number. For a typical message with byte-oriented operands only, the appearance within the information field of an HDLC frame, i.e., in order of transmission, is shown in figure 6-1. A typical message including a non-byte-oriented operand (16-bit number) is shown in figure 6-2.

Where it is necessary to reference the parameters of the Command and Status messages by number, the parameters are considered to be numbered in order of transmission, beginning with number one. (Note that the parameter numbering does not necessarily correspond to the byte numbering.) Parameter numbering is not shown in the following format descriptions.

All spare bytes and spare parameter values are reserved and can be assigned only through amendment to this generic specification.

6.3.2 PSP Command messages

PSPCONFIGURATION (01)

PARAMETERS Command (1) Data (1)

ORIGINATOR PSP

FUNCTION To inform the IOT of the PSP configuration.

TRANSMISSION ORDER:

	ID CODE	Command	Data
Byte Nos.	B0	B1	B2

DEFINITION:

```
PSPCONFIGURATION                 ::= IDCODE  Command  Data
  IDCODE                         ::= B0  --byte 0
    B0                           ::= 01 hex
  Command  Data                  ::= B1  B2
    B1  B2                       ::= VERIFYOUTPUTDELIVERY
                                       ReturnOutputSheetNumber
                                     | VERIFYDUPLEXDELIVERY
                                       ReturnDuplexSheetNumber
                                     | SCHEDULINGOFFSET  NumOfPageSyncs
                                     | DATALINKACKTIME  Time
                                     | RETURNIOTCONFIGURATION  NoData
                                     | SPARECOMMANDS  SpareData
      VERIFYOUTPUTDELIVERY       ::= 01 hex     -- Return the sheet number of
                                                delivered sheets as requested.
        ReturnOutputSheetNumber  ::= No | OfLastSheet | OfEachSheet
                            No   ::= 00 hex
                   OfLastSheet   ::= 01 hex
                   OfEachSheet   ::= 02 hex    -- Refer to "Appl. notes," following.
      VERIFYDUPLEXDELIVERY       ::= 02 hex    -- Return the sheet number of each
                                               sheet delivered to the duplex tray, as
                                               requested.
        ReturnDuplexSheetNumber  ::= No | Yes
                            No   ::= 00 hex
                           Yes   ::= 01 hex
```

SCHEDULINGOFFSET	::=	03 hex	-- The number of Page Sync positive transitions between PSPPRINT and IOTVIDEOREQUEST as required by the PSP.
Undefined	::=	00 hex	
NumOfPageSyncs	::=	[01 .. FF] hex	
DATALINKACKTIME	::=	04 hex	
Time	::=	[00 .. FF] hex	-- in milliseconds
RETURNIOTCONFIGURATION	::=	05 hex	
NoData	::=	00 hex	-- data not applicable
SPARECOMMANDS	::=	[06 .. FF] hex	
SpareData	::=	[00 .. FF] hex	

RESPONSE

IOTCONFIGURATION in response to RETURNIOTCONFIGURATION command parameter, otherwise none

APPLICATION NOTES

Where sheet number is required, it is to be understood that copy number must also be included.

DATALINKACKTIME is the time which the PSP guarantees not to exceed when acknowledging (positively or negatively) a received data link frame which requires acknowledgement. This time becomes the acknowledgement time-out employed by the IOT data link transmitting function. Refer to section 5.3.9-e.

If ReturnOutputSheetNumber-OfEachSheet is designated in conjunction with an output finishing device that has an intermediate destination, an 8x hex series destination code will be returned as each sheet is delivered to the intermediate destination. When the set is ejected to the final destination, an 0x hex series final destination code will be returned with the last sheet number only. (Refer to IotSheetDelivered.)

PSPMETARESET (02)

PARAMETERS PspMessage (1)

ORIGINATOR PSP

FUNCTION To inform the IOT that an unrecoverable problem has been detected and instruct the IOT to take appropriate action.

TRANSMISSION ORDER:

	ID CODE	PspMessage
Byte Nos.	B0	B1

DEFINITION:

PSPMETARESET	::=	IDCODE PspMessage
IDCODE	::=	B0 --byte 0
B0	::=	02 hex
PspMessage	::=	B1
B1	::=	PspPowerOffNotice | PspSoftwareReset | Spare
PspPowerOffNotice	::=	01 hex -- Notice that IOT power loss will occur.
PspSoftwareReset	::=	02 hex -- IOT to reset w/o softload.
Spare	::=	[03 .. FF] hex -- Program specific.

RESPONSE none

APPLICATION NOTES This message may be sent in the Un-numbered information format of the HDLC frame to preclude the possibility of invoking the data link time-out and retransmission strategy during implied exception conditons.

PSPNEXTBANKREQUEST (03)

PARAMETERS PlateMode (1) SheetNumber (2) NumberOfCopies(2) NextBankTaskInfoA (1) NextBankTaskInfoB (1) NextBankTaskInfoC (1) BindexerFillLimit (1) SorterBinDestination (1) StitchAPosition (1) StitchBPosition (1) FeederPermissions (1) DestinationPermissions (1) ManualFeederPermissions (1) ManualDestinationPermissions (1) PaperWidth (2) PaperLength (2) PaperType (1) FutureFinishingOptions (2) JobNumber (1) ContrastControl (1) ContrastData (2)

ORIGINATOR PSP

FUNCTION To inform the IOT about a bank that will be run. The switch to the next bank will happen after the PSP requests the SheetNumber via PSPPRINT.

TRANSMISSION ORDER:

	ID Code	Plate Mode	SheetNumber		NumberOfCopies		NextBankTaskInfo A	B	C	BinFill Limit	SorterBin Destin.
Byte Nos.	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10

StitchA Position	StitchB Position	Permissions Feeder	Destin.	ManualPermiss'ns Feeder	Destin.	Paper Type.	PaperWidth		PaperLength	
B11	B12	B13	B14	B15	B16	B17	B18	B19	B20	B21

FutureFinishing Options		Job Number	Contrast Control	ContrastData	
B22	B23	B24	B25	B26	B27

DEFINITION:

PSPNEXTBANKREQUEST ::= IDCODE PlateMode SheetNumber NumberOfCopies NextBankTaskInfoA NextBankTaskInfoB NextBankTaskInfoC BindexerFillLimit SorterBinDestination StitchAPosition StitchBPosition FeederPermissions DestinationPermissions ManualFeederPermissions ManualDestinationPermissions PaperWidth PaperLength PaperType FutureFinishingOptions JobNumber ContrastAction ContrastData

IDCODE ::= B0 --byte 0

B0 ::= 03 hex

PlateMode ::= B1

B1 ::= PageMode ColorMode ResolutionMode

PageMode ::= d1 d0

d1 d0 ::= Duplex | Simplex | SimultaneousDuplex | Spare

Duplex ::= 00 binary
Simplex ::= 01 binary
SimultaneousDuplex ::= 10 binary
Spare ::= 11 binary

```
ColorMode                          ::=  d5 d4 d3 d2
            PlateDeadCycle         ::=  000000 binary
      d2                           ::=  NotColor0 | Color0
            NotColor0              ::=  0 binary
            Color0                 ::=  1 binary
      d3                           ::=  NotColor1 | Color1
            NotColor1              ::=  0 binary
            Color1                 ::=  1 binary
      d4                           ::=  NotColor2 | Color2
            NotColor2              ::=  0 binary
            Color2                 ::=  1 binary
      d5                           ::=  NotColor3 | Color3
            NotColor3              ::=  0 binary
            Color3                 ::=  1 binary
  ResolutionMode                   ::=  d7 d6
      d6                           ::=  SimplexRes1 | SimplexRes2
            SimplexRes1            ::=  0 binary -- IOT specific
            SimplexRes2            ::=  1 binary -- IOT specific
      d7                           ::=  DuplexRes1 | DuplexRes2
            DuplexRes1             ::=  0 binary -- IOT specific
            DuplexRes2             ::=  1 binary -- IOT specific
SheetNumber                        ::=  B2 B3
  B2 B3                            ::=  Undefined | StartingSheetNumber
            Undefined              ::=  0000 hex  -- never used
            StartingSheetNumber    ::=  [0001 .. FFFF] hex
NumberOfCopies                     ::=  B4 B5
  B4 B5                            ::=  Undefined | NumberOfCopies
            Undefined              ::=  0000 Hex
            NumberOfCopies         ::=  [0001 .. FFFF] hex
NextBankTaskInfoA                  ::=  B6
  B6                               ::=  GoodSheetDestination  Feeder
                                        TaskFinishing
  GoodSheetDestination             ::=  d2 d1 d0
      d2 d1 d0                     ::=  Destination0 | Destination1
                                        | Destination2 | Destination3
                                        | Destination4 | Destination5
                                        | Destination6 | Destination7
            Destination0           ::=  000 binary
            Destination1           ::=  001 binary
            Destination2           ::=  010 binary
            Destination3           ::=  011 binary
            Destination4           ::=  100 binary
            Destination5           ::=  101 binary
            Destination6           ::=  110 binary
            Destination7           ::=  111 binary
```

Feeder	::=	d5 d4 d3
d5 d4 d3	::=	Feeder0 | Feeder1 | Feeder2 | Feeder3 | Feeder4 | Feeder5 | Feeder6 | Feeder7
Feeder0	::=	000 binary
Feeder1	::=	001 binary
Feeder2	::=	010 binary
Feeder3	::=	011 binary
Feeder4	::=	100 binary
Feeder5	::=	101 binary
Feeder6	::=	110 binary
Feeder7	::=	111 binary
TaskFinishing	::=	d7 d6
d6	::=	Collated | Uncollated
Collated	::=	0 binary
Uncollated	::=	1 binary
d7	::=	NToOne | OneToN
NToOne	::=	0 binary
OneToN	::=	1 binary
NextBankTaskInfoB	::=	B7
B7	::=	ScratchSheetDestination StartOfJob EndOfJob InterruptJobAsap ResumeInterruptedJob Spare
ScratchSheetDestination	::=	d2 d1 d0
d2 d1 d0	::=	Destination0 | Destination1 | Destination2 | Destination3 | Destination4 | Destination5 | Destination6 | Destination7
Destination0	::=	000 binary
Destination1	::=	001 binary
Destination2	::=	010 binary
Destination3	::=	011 binary
Destination4	::=	100 binary
Destination5	:=	101 binary
Destination6	::=	110 binary
Destination7	::=	111 binary
StartOfJob	::=	d3 --This bank begins new IOT job.
d3	::=	No | Yes
No	::=	0 binary
Yes	::=	1 binary
EndOfJob	::=	d4 -- If last job, start cycle down after StartingSheetNumber is successfully imaged.
d4	::=	No | Yes
No	::=	0 binary
Yes	::=	1 binary

InterruptJobAsap	::=	d5 --This is the 1st bank of an ASAP interrupt job
d5	::=	No | Yes
No	::=	0 binary
Yes	::=	1 binary
ResumeInterruptedJob	::=	d6 --Return to the banks indicated by JobNumber in this message.
d6	::=	No | Yes
No	::=	0 binary
Yes	::=	1 binary
Spare	::=	d7
d7	::=	0 binary
NextBankTaskInfoC	::=	B8
B8	::=	OffsetOptions StichingOptions BindingOptions UnloadOptions
OffsetOptions	::=	d2 d1 d0
d1 d0	::=	NoOffset | Bank | Set | Bin
NoOffset	::=	00 binary
Bank	::=	01 binary
Set	::=	10 binary
Bin	::=	11 binary
d2	::=	NoRecoveryOffSet | RecoveryOffSet
NoRecoveryOffSet	::=	0 binary -- Specifies whether one sheet is to be offset at the point of recovery.
RecoveryOffSet	::=	1 binary
StitchingOptions	::=	d4 d3
d4 d3	::=	None | PortraitSingleStitch | LandscapeSingleStitch | DoubleStitch
None	::=	00 binary
PortraitSingleStitch	::=	01 binary
LandscapeSingleStitch	::=	10 binary
DoubleStitch	::=	11 binary
BindingOptions	::=	d6d5
d6d5	::=	None | BindOnly | DoubleStitchAndBind | Spare
None	::=	00 binary
BindOnly	::=	01 binary
DoubleStitchAndBind	::=	10 binary
Spare	::=	11 binary
UnloadOptions	::=	d7
d7	::=	UnloadAtJobComplete | UnloadAtMaxCapacity
UnloadAtJobComplete	::=	0 binary
UnloadAtMaxCapacity	::=	1 binary

```
BinFillLimit                                  ::=  B9
    B9                                        ::=  PhysicalLimit | ProgrammedLimit
                  PhysicalLimit               ::=  00 hex
                  ProgrammedLimit             ::=  [01 .. FF] hex -- Limits and units of
                                                        measurement are
                                                        destination specific.
SorterBinDestination                          ::=  B10
    B10                                            ::=  Undefined | SorterBinNumber
                  Undefined                   ::=  00 hex
                  SorterBinNumber             ::=  [01 .. FF] hex -- Applicable if a
                                                        multi-bin sorter has been
                                                        specified under
                                                        GoodSheetDestination
StitchAPosition                               ::=  B11
    B11                                       ::=  Position
                  Position                    ::=  [00 .. FF] hex  -- IOTspecific
StitchBPosition                               ::=  B12
    B12                                       ::=  Position
                  Position                    ::=  [00 .. FF] hex  -- IOT specific
FeederPermissions                             ::=  B13
    B13                                       ::=  FeederIdentification  -- Defined in
                                                             NextBankTaskInfoA
              FeederIdentification            ::=  [d7 .. d0]
                d0                            ::=  Feeder0
                d1                            ::=  Feeder1
                d2                            ::=  Feeder2
                d3                            ::=  Feeder3
                d4                            ::=  Feeder4
                d5                            ::=  Feeder5
                d6                            ::=  Feeder6
                d7                            ::=  Feeder7
                  Feeder0 .. Feeder7          ::=    SwitchNotPermitted |
                                                     SwitchPermitted
                  SwitchNotPermitted          ::=  0  binary
                  SwitchPermitted             ::=  1  binary
DestinationPermissions                        ::=  B14
    B14                                       ::=  DestinationIdentification  -- Defined
                                                             in NextBankTaskInfoA
              DestinationIdentification       ::=  [d7 .. d0]
                d0                            ::=  Destination0
                d1                            ::=  Destination1
                d2                            ::=  Destination2
                d3                            ::=  Destination3
                d4                            ::=  Destination4
                d5                            ::=  Destination5
                d6                            ::=  Destination6
                d7                            ::=  Destination7
```

Destination0 .. Destination7	::=	SwitchNotPermitted | SwitchPermitted
SwitchNotPermitted	::=	0 binary
SwitchPermitted	::=	1 binary
ManualFeederPermissions	::=	B15
B15	::=	FeederIdentification -- Defined in NextBankTaskInfoA
FeederIdentification	::=	[d7 .. do]
d0	::=	Feeder0
d1	::=	Feeder1
d2	::=	Feeder2
d3	::=	Feeder3
d4	::=	Feeder4
d5	::=	Feeder5
d6	::=	Feeder6
d7	::=	Feeder7
Feeder0 .. Feeder7	::=	SwitchNotPermitted | SwitchPermitted
SwitchNotPermitted	::=	0 binary
SwitchPermitted	::=	1 binary
ManualDestinationPermissions	::=	B16
B16	::=	DestinationIdentification -- Defined in NextBankTaskInfoA
DestinationIdentification	::=	[d7 .. d0]
d0	::=	Destination0
d1	::=	Destination1
d2	::=	Destination2
d3	::=	Destination3
d4	::=	Destination4
d5	::=	Destination5
d6	::=	Destination6
d7	::=	Destination7
Destination0 .. Destination7	::=	SwitchNotPermitted | SwitchPermitted
SwitchNotPermitted	::=	0 binary
SwitchPermitted	::=	1 binary
PaperType	::=	B17
B17	::=	d7d6d5d4d3d2d1d0
d0	::=	NotSpecified | Specified
NotSpecified	::=	0 -- default to contents of specified tray
Specified	::=	1 -- as below
d1	::=	NotTransparency | Transparency
NotTransparency	::=	0
Transparency	::=	1

XEROX Private Data

d2	::=	NotDrilled | Drilled
NotDrilled	::=	0
Drilled	::=	1
d7d6d5d4d3	::=	NotTabStock | NumberOfTabs
NotTabStock	::=	00000 binary
NumberOfTabs	::=	[00001 .. 11111] binary -- 00001 is full tab, 00010 is 2 half-length tabs, etc.
PaperWidth	::=	B18B19
B18B19	::=	Default | WidthDimension
Default	::=	0000 Hex -- Whatever is in specified tray.
WidthDimension	::=	[0001 .. FFFF] hex -- PaperWidth in mm.
PaperLength	::=	B20B21
B20B21	::=	Default | LengthDimension
Default	::=	0000 Hex -- Whatever is in specified tray.
LengthDimension	::=	[0001 .. FFFF] hex -- PaperLength in mm.
FutureFinishingOptions	::=	B22B23
B22B23	::=	CopySensitive InhibitBindexerMode Spares
CopySensitive	::=	d0 -- Printed data on sheets may differ from copy to copy.
d0	::=	No | Yes
No	::=	0 binary
Yes	::=	1 binary
InhibitBindexerMode	::=	d1 -- inhibit use of bindexer algorithms.
d1	::=	No | Yes
No	::=	0 binary
Yes	::=	1 binary
Spares	::=	[d15 .. d2]
JobNumber	::=	B24
B24	::=	Undefined | JobIdentifier
Undefined	::=	00 hex
JobIdentifier	::=	[01 .. FF] hex
ContrastControl	::=	B25
B25	::=	Normal | Darker | Lighter | Programmable | Spare
Normal	::=	00 hex
Darker	::=	01
Lighter	::=	02
Programmable	::=	03 -- refer to ContrastData
Spare	::=	[04 .. FF] hex

ContrastData	::=	B26B27
B26B27	::=	ContrastSetting
ContrastSetting	::=	[0000 .. FFFF] hex -- definition is IOT specific

RESPONSE none

APPLICATION NOTES

Refer to sections 6.4 and 6.5 for timing requirements, numbering and sequencing conventions, and examples of this message in operating sequences.

CopyNumber is not included in this command, as jobs will always be programmed on set boundries.

The convention for assigning color designations under *PlateMode/ColorMode* is as follows: For single color printing, usually black, Color0 shall designate the single color. For single hi-lite color printing, Color0 shall designate the principal color, usually black, and Color1 shall designate the hi-lite color. (The hue of the hi-lite color is IOT specific.) For multicolor hi-lite color printing, Color0 shall designate the principal color, usually black, and Color1 through Color3 shall designate IOT specific hi-lite colors. For full color printing, Color0 through Color3 are black, magenta, cyan, and yellow, respectively.

OffsetOptions provides for offsetting sheets, sets, stacks, or parts of stacks in either collated or uncollated mode. *Bank* specifies that the sheets of the current bank are to be offset from the sheets of the previous bank. If this mode is selected in the first bank of a collated job with a stacker output, the copy sets will be offset in the stacker. It may also be used in this case to offset the sheets of any bank within the job. *Set* specifies to a bindexer that the collated sets are to be offset when ejected to the output stacker. *Bin* specifies that the uncollated contents of a bindexer bin are to be offset when ejected to the output stacker, and the number of sheets to be accumulated before ejecting is given in B9, *BinFillLimit*.

BinFillLimit may apply to any destination. The limits and the units of measurement are destination specific. If destination switching is permitted (byte 14 or 16) switching must occur only when the specified limit has been reached. Because the sensors for the programmed limits may be different in different destinations, the BinFillLimit for the new destination should automatically default to the PhysicalLimit. Optionally, if the IOT has the capability, it may set the limit to the nearest equivalent of the original BinFillLimit specification.

StitchingOptions provides for a single stitch (staple) in the upper left corner of either a portrait mode document, stitch A position, or of a landscape mode document, stitch B position. These stitch positions are both along the slow scan direction, which is normally the long edge of a sheet. Therefore, the options also provide for double stitching along the long edge of either a portrait or landscape mode document. The default stitch positions are IOT specific, but offsets from these locations are programmable via B11 *StitchAPosition* and B12 *StitchBPosition*. The default positions are the outermost positions, and each offset moves the respective stitch location inboard along the long edge. Thus, symmetrically located double stitches are produced by specifying identical offsets for *StitchAPosition and StitchBPosition*.

Provision is made for the PSP to designate its expectations regarding the output medium which the IOT will utilize, i.e., size and type. The length and width parameters for paper size are each assigned fields two bytes long. This provides ample range for any U.S., ISO, or JIS paper size. Note that paper width is defined as the dimension in the slow-scan direction, and length is defined as the dimension in the fast-scan direction, regardless of whether the actual paper is fed in the portrait or landscape orientation.

If the IOT detects a discrepancy between the paper characteristics specified in this message and the characteristics of the paper actually being fed, the IOT must report a fault. If detection of a parameter is not implemented in the IOT, it should report NotAvailable in IotOperationalInfo and ignore the fault reporting requirement. If a parameter is only detectable when paper is being fed, the IOT may change its IotOperationalInfo response and fault reporting stance after sufficient paper has been fed. Note that the IOT is not required to report PaperType in IotOperationalInfo; but if the type parameters are detectable, it must report a fault, if a discrepancy is found. PaperType is also intended for use by the IOT, as necessary, to condition its paper path to handle special media.

Specifying a change in paper size may result in an IOT configuration change (refer to IotConfiguration). This may entail a delay, which would be managed by the IOT through IotStateInfo and IotOperationalInfo messages.

The dimensions of the operational Standard Image Frame are determined from the Paper Size/Standard Image Frame matrix (refer to IotConfiguration) and the paper size specified in this message.

In a gapless printer (web feed), the paper width specified in this message determines the dimension of page sync to be generated by the IOT.

InterruptJobASAP indicates the first bank of a job which must be allowed to interrupt a job in process as soon as possible. This will typically be on a sheet boundary, but may be on a set-boundary if that is the first opportunity available to the IOT. In order to avoid mixing the output of the interrupting job with the output of the interrupted job, the PSP should, if possible, specifiy a destination for the interrupting job which is different from the destination for the interrupted job. If a different destination compatible with the output of the interrupting job is not available, the PSP should at least offset the interrupting job from the interrupted job.

ResumeInterruptedJob notifies the IOT that it should resume the interrupted job designated by the JobNumber. All other parameters of the message are to be ignored. (Also refer to section 6.4.6, "Job Interrupt Protocol").

CopySensitive indicates that the printed data on sheets may differ from copy to copy and informs the IOT that the bindexer minimum-travel algorithms should not be used for sheets beginning with the designated *SheetNumber*.

InhibitBindexerMode indicates that the bindexer mode of requesting multiple copies of each sheet should not be used by the IOT. This is specified if the PSP knows that the number of pages in the job exceeds the capacity of the bindexer.

PSPPRINT (04)

PARAMETERS PlateNumber (1) SheetNumber (2) CopyNumber (2) JobNumber (1)

ORIGINATOR PSP

FUNCTION To confirm that the PSP intends to deliver the video for the sheet number, plate number, and copy number specified, usually this is confirmation of the IOT's previous IotVideoHint.

TRANSMISSION ORDER:

	ID CODE	Plate Number	SheetNumber		CopyNumber		Job Number
Byte Nos.	B0	B1	B2	B3	B4	B5	B6

DEFINITION:

PSPPRINT	::= IDCODE PlateNumber SheetNumber CopyNumber JobNumber
IDCODE	::= B0 --byte 0
B0	::= 04 hex
PlateNumber	::= B1
B1	::= PageMode PlateColor PlateResolution
PageMode	::= d1 d0
d1 d0	::= Duplex | Simplex | SimultaneousDuplex | Spare1
Duplex	::= 00 binary
Simplex	::= 01 binary
SimultaneousDuplex	::= 10 binary
Spare1	::= 11 binary
PlateColor	::= d5 d4 d3 d2
PlateDeadCycle	::= 000000 binary
d2	::= NotColor0 | Color0
NotColor0	::= 0 binary
Color0	::= 1 binary
d3	::= NotColor1 | Color1
NotColor1	::= 0 binary
Color1	::= 1 binary
d4	::= NotColor2 | Color2
NotColor2	::= 0 binary
Color2	::= 1 binary
d5	::= NotColor3 | Color3
NotColor3	::= 0 binary
Color3	::= 1 binary
ResolutionMode	::= d7 d6
d6	::= SimplexRes1 | SimplexRes2
SimplexRes1	::= 0 binary -- IOT specific
SimplexRes2	::= 1 binary -- IOT specific
d7	::= DuplexRes1 | DuplexRes2
DuplexRes1	::= 0 binary -- IOT specific

DuplexRes2	::=	1 binary -- IOT specific
SheetNumber	::=	B2 B3
B2 B3	::=	SheetDeadcycle | SheetIdentifier0
SheetDeadCycle	::=	0000 hex
SheetIdentifier	::=	[0001 .. FFFF] hex
CopyNumber	::=	B4 B5
B4 B5	::=	SampleCopy | CopyIdentifier
SampleCopy	::=	0000 hex
CopyIdentifier	::=	[0001 .. FFFF] hex
JobNumber	::=	B6
B6	::=	Undefined | JobIdentifier
Undefined	::=	00 hex
JobIdentifier	::=	[01 .. FF] hex

RESPONSE IOTVIDEOREQUEST PlateNumber SheetNumber CopyNumber JobNumber

APPLICATION NOTES

Refer to sections 6.4 and 6.5 for timing requirements, numbering and sequencing conventions, and examples of this message in operating sequences.

The convention for assigning color designations under PlateNumber/PlateColor is as follows: For single color printing, usually black, Color0 shall designate the single color. For single hi-lite color printing, Color0 shall designate the principal color, usually black; and Color1 shall designate the hi-lite color (the hue of the hi-lite color is IOT specific). For multicolor hi-lite color printing, Color0 shall designate the principal color, usually black; and Color1 through Color3 shall designate IOT specific hi-lite colors. For full color printing, Color0 through Color3 are black, magenta, cyan, and yellow, respectively.

PSPREADIOTMEMORY (06)

PARAMETERS Spare(1) StartAddress (4) EndAddress (4)

ORIGINATOR PSP

FUNCTION To request a segment of IOT memory to be sent to the PSP in a series of IOTWRITEMEMORYTOPSP messages.

TRANSMISSION ORDER:

	ID CODE	Spare	StartAddress				EndAddress			
Byte Nos.	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9

DEFINITION:

PSPREADIOTMEMORY	::= IDCODE Spare StartAdress EndAddress
IDCODE	::= B0 --byte 0
B0	::= 06 hex
Spare	::= B1
B1	::= [00 .. FF] hex
StartAddress	::= B2B3B4B5
B2B3B4B5	::= [00000000 .. FFFFFFFF] hex
EndAddress	::= B6B7B8B9
B6B7B8B9	::= [00000000 .. FFFFFFFF] hex

RESPONSE IOTWRITEMEMORYTOPSP EndOfBlock StartAddress Length Data

APPLICATION NOTES This message is used by the PSP to initiate a memory upload from the IOT.

PSPREADIOTSTATE (07)

PARAMETERS (None)

ORIGINATOR PSP

FUNCTION To request the IOT state be sent to the PSP.

TRANSMISSION ORDER:

	ID CODE
Byte Nos.	B0

DEFINITION:

PSPREADIOTSTATE	::= IDCODE
IDCODE	::= B0 --byte 0
B0	::= 07 hex

RESPONSE IOTSTATEINFO State

PSPREADIOTOPERATIONALINFO (08)

PARAMETERS InformationType (1)

ORIGINATOR PSP

FUNCTION Allows the PSP to determine the status of the IOT at any time.

TRANSMISSION ORDER:

	ID CODE	Information Type
Byte Nos.	B0	B1

DEFINITION:

PSPREADIOTSTATUS ::= IDCODE InformationType

IDCODE ::= B0 --byte 0

B0 ::= 08 hex

InformationType ::= B1

B1 ::= Undefined | TECHREPCLEARONLYFAULTS
| OPERATORCLEARFAULTS | TECHREPRETRYFAULTS
| HINTMESSAGE | INFOMESSAGE | FEEDER0STATUS
| FEEDER1STATUS | FEEDER2STATUS |
| FEEDER3STATUS
| FEEDER4STATUS | FEEDER5STATUS |
| FEEDER6STATUS
| FEEDER7STATUS | DESTINATION0STATUS |
| DESTINATION1STATUS
| DESTINATION2STATUS | DESTINATION3STATUS
| DESTINATION4STATUS | DESTINATION5STATUS
| DESTINATION6STATUS | DESTINATION7STATUS
| CRASHRECOVERYSTATUS | Spare

UNDEFINED ::= 00 hex
TECHREPCLEARONLYFAULTS ::= 01 hex
OPERATORCLEARFAULTS ::= 02 hex
TECHREPRETRYFAULTS ::= 03 hex
HINTMESSAGE ::= 04 hex
INFOMESSAGE ::= 05 hex
FEEDER0STATUS ::= 06 hex
FEEDER1STATUS ::= 07 hex
FEEDER2STATUS ::= 08 hex
FEEDER3STATUS ::= 09 hex
FEEDER4STATUS ::= 0A hex
FEEDER5STATUS ::= 0B hex
FEEDER6STATUS ::= 0C hex
FEEDER7STATUS ::= 0D hex
DESTINATION0STATUS ::= 0E hex
DESTINATION1STATUS ::= 0F hex
DESTINATION2STATUS ::= 10 hex
DESTINATION3STATUS ::= 11 hex
DESTINATION4STATUS ::= 12 hex
DESTINATION5STATUS ::= 13 hex
DESTINATION6STATUS ::= 14 hex
DESTINATION7STATUS ::= 15 hex
CRASHRECOVERYSTATUS ::= 16 hex
SPARE ::= [17 .. FF] hex

RESPONSE IOTOPERATIONALINFO InformationType DataWord

PSPREQUESTIOTDIAGNOSTIC (09)

PARAMETERS Command (1) Utility (2) ParameterID (2) ParameterValue(2)

ORIGINATOR PSP

FUNCTION To invoke predetermined IOT diagnostic utility programs from the PSP.

TRANSMISSION ORDER:

	ID Code	Command	Utility		ParameterID		ParameterValue	
Byte Nos.	B0	B1	B2	B3	B4	B5	B6	B7

DEFINITION:

PSPREQUESTIOTDIAGNOSTIC	::= IDCODE Command Utility ParameterID ParameterValue
IDCODE	::= B0 --byte 0
B0	::= 09 hex
Command	::= B1
B1	::= Enter | Start | Stop | Exit | Read | Write | Spare
Enter	::= 00 hex
Start	::= 01 hex
Stop	::= 02 hex
Exit	::= 03 hex
Read	::= 04 hex
Write	::= 05 hex
Spare	::= [06 .. FF] hex
Utility	::= B2B3
B2B3	::= [00..FF] -- ID of IOT specific diagnostic utility.
ParameterID	::= B4B5
B4B5	::= [0000..FFFF] -- Set of parameters unique to Utility and Command.
ParameterValue	::= B6B7
B6B7	::= [0000..FFFF] -- Set of parameter values.

RESPONSE IOTDIAGNOSTICRESPONSE Status Utility ParameterID ParameterValue

APPLICATION NOTES

This command is valid at any time that the requested Utility can be run. The Utility invoked may perform certain diagnostics concurrently with the processing of user jobs, or the utility may supply the banks to run a diagnostic job. If the utility cannot be run because of priority activity in the IOT, this message shall be rejected via the status field of IotDiagnosticResponse (for example, a diagnostic job will not be run if the request is made while a user's job is in process).

Enter is the command to enter the Utility program. Activity may or may not occur on entering the Utility.

Start is the command to start a Utility program which has been entered without starting, or to restart a Utility program which has been stopped.

Stop is the command to stop a Utility which is running, without exiting.

Exit is the command to leave the Utility program. If *Exit* is sent while a Utility is running, the Utility will stop and then exit.

Read is a request for the value of the parameter listed in ParameterID. (The value may also be the state of a device or the contents of a memory location.)

Write is a command to change the value of the parameter listed in ParameterID according to the contents of ParameterValue. (*Write* may also change the state of a device or the contents of a memory location.)

When not applicable, the ParameterID and/or ParameterValue fields should be 0000 Hex.

Every Command used in byte 6 requires a response. (Refer to IotDiagnosticResponse.)

Utility numbering must be in the range 1 to 999 decimal in accordance with the *Multi-National Standard for Diagnostic ProgramNumbers and Status Codes* (#7000P02860).

Also refer to section 6.6.

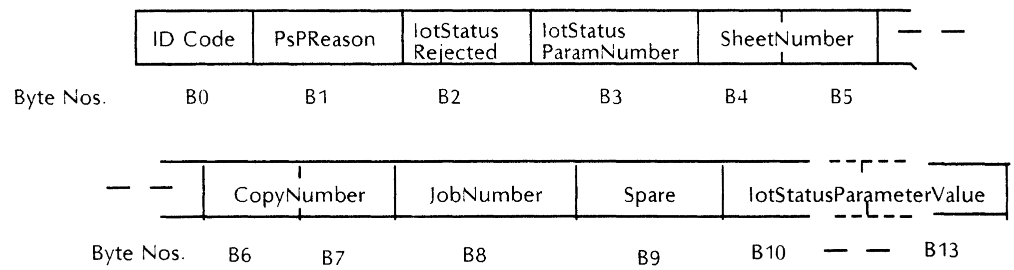
PSPREJECTIOTMESSAGE (0B)

PARAMETERS PspReason (1) IotStatusRejected (1) IotStatusParameterNumber (1) SheetNumber (2) CopyNumber (2) JobNumber (1) Spare (1) IotStatusParameterValue (variable 1 to 4)

ORIGINATOR PSP

FUNCTION To inform the IOT that the status sent to the PSP is not valid at the time of receipt.

TRANSMISSION ORDER:



DEFINITION:

PSPREJECTIOTMESSAGE ::= IDCODE PspReason IotStatusRejected IotStatusParameterNumber SheetNumber CopyNumber JobNumber Spare IotStatusParameterValue

IDCODE ::= B0 --byte 0

B0 ::= 0B hex

PspReason ::= B1

B1 ::= UNDEFINEDSTATUS | ILLEGALSEQUENCE | INVALIDSTATUSPARAMETER | STATUSBUFFERFULL | UNSUPPORTEDSTATUS | INVALIDFORMAT | Spare

UNDEFINEDSTATUS ::= 01 hex -- Status ID Code not recognized as valid.

ILLEGALSEQUENCE ::= 02 hex -- Receipt of valid status at an invalid PSP state for that status.

INVALIDSTATUSPARAMETER ::= 03 hex -- IOT Status Parameter No. out of valid range.

STATUSBUFFERFULL ::= 04 hex -- PSP too busy to handle any further status at this time.

UNSUPPORTEDSTATUS ::= 05 hex -- Status ID Code not implemented in this revision software.

INVALIDFORMAT ::= 06 hex -- too few or too many parameters for the valid ID code received.

Spare ::= [07 .. FF] hex

IotStatusRejected	:: =	B2	
B2	:: =	IOTCONFIGURATION | IOTVIDEOHINT | IOTVIDEOREQUEST | IOTWRITEMEMORYTOPSP | IOTSTATEINFO | IOTOPERATIONALINFO | IOTDIAGNOSTICRESPONSE | IOTSHEETDELIVERED | IOTREQUESTMEMORY | IOTSWITCHINFO	
IOTCONFIGURATION	:: =	81 hex	
IOTVIDEOHINT	:: =	83 hex	
IOTVIDEOREQUEST	:: =	84 hex	
IOTWRITEMEMORYTOPSP	:: =	86 hex	
IOTSTATEINFO	:: =	87 hex	
IOTOPERATIONALINFO	:: =	88 hex	
IOTDIAGNOSTICRESPONSE	:: =	89 hex	
IOTSHEETDELIVERED	:: =	8C hex	
IOTREQUESTMEMORY	:: =	8D hex	
IOTSWITCHINFO	:: =	8E hex	
IotStatusParameterNumber	:: =	B3	
B3	:: =	[00 .. FF] hex	-- The number of the first status parameter sent by the IOT which was found to be in error; numbered in order of transmission, beginning with # 1.
SheetNumber	:: =	B4 B5	
B4 B5	:: =	NotApplicable | SheetIdentifer	
NotApplicable	:: =	0000	
SheetIdentifier	:: =	[0001 .. FFFF] hex	
CopyNumber	:: =	B6 B7	
B6 B7	:: =	NotApplicable | CopyIdentifier	
NotApplicable	:: =	0000	
CopyIdentifier	:: =	[0001 .. FFFF] hex	
JobNumber	:: =	B8	
B8	:: =	NotApplicable | JobIdentifier	
NotApplicable	:: =	0000	
JobIdentifier	:: =	[01 .. FF] hex	
Spare	:: =	B9	
B9	:: =	[00 .. FF] hex	
IotStatusParameterValue	:: =	[B10 .. B13]	
[B10 .. B13]	:: =	[00 .. FFFFFFFF] hex -- Status parameter value sent by the IOT.	

RESPONSE	IOTSTATEINFO	MachineState	Substates
	IOTOPERATIONALINFO	InformationType	DataWord

APPLICATION NOTES

When *Invalid Format* is specified, the entry for IotStatusParameterNumber should be as follows:

> If there are too few parameters, enter the number of the first missing parameter, i.e., the actual number of parameters received plus one.
>
> If there are too many parameters, enter the number of the first excess parameter, i.e., the expected number plus one.
>
> (Refer to section 6.3.1 for convention on parameter numbering.)

SheetNumber, CopyNumber, and JobNumber should be specified as NotApplicable if they were not included in the message being rejected.

The IotStatusParameterValue field is variable, according to the length of the specific parameter value to be transmitted.

PspSheetBankAbort (0C)

PARAMETERS AbortType (1) SheetNumber (2) CopyNumber (2) JobNumber (1)

ORIGINATOR PSP

FUNCTION To abort the sheet designated by the SheetNumber and CopyNumber, and/or the current bank or all banks.

TRANSMISSION ORDER:

	ID CODE	Abort Type	SheetNumber		CopyNumber		Job Number
Byte Nos.	B0	B1	B2	B3	B4	B5	B6

DEFINITION:

PspSheetBankAbort ::= IDCode AbortType SheetNumber CopyNumber JobNumber

IDCode ::= B0 --byte 0

B0 ::= 0C hex

AbortType ::= B1

B1 ::= SheetAbortA | SheetAbortB | JobAbortWithRecovery | JobAbortWithoutRecovery | AllJobsAbort | Spare

SheetAbortA ::= 00 hex -- Abort the specified sheet only. The video may be damaged.

SheetAbortB ::= 01 hex -- Abort the specified sheet only. The video is guaranteed white.

JobAbortWithRecovery ::= 02 hex -- if output jam occurs, attempt recovery of sheets preceding specified SheetNumber/CopyNumber.

JobAbortWithoutRecovery ::= 03 hex -- if output jam occurs, do not attempt recovery of sheets preceding specified SheetNumber/CopyNumber.

AllJobsAbort ::= 04 hex -- Cancel all banks of all jobs and cycledown the IOT.

Spare ::= [05 .. FF] hex

SheetNumber ::= B2 B3

B2 B3 ::= Undefined | Specific | Sheet Number

Undefined ::= 0000 hex

SpecifiedSheetNumber ::= [0001 ..FFFF] hex --Together with CopyNumber, designates sheet to be skipped, sheet to try again, or termination point of the bank.

CopyNumber ::= B4 B5

B4 B5 ::= Undefined | SpecifiedCopyNumber

Undefined ::= 0000 hex

SpecifiedCopyNumber ::= [0001 ..FFFF] hex -- See comment under SheetNumber.

JobNumber ::= B6

B6 ::= AllJobs | SpecifiedJobNumber

AllJobs	::= 0000 hex -- used with AllJobsAbort only.
SpecifiedJobNumber	::= [0001 ..FFFF] hex

RESPONSE IOTSTATEINFO MachineState Substates

APPLICATION NOTES

In the event of a failure to image properly, the specified sheet may be aborted with SheetAbortA or SheetAbortB. In either case, this sheet will be delivered to the scratch sheet destination previously specified by a PspNextBankRequest. The job will continue with the same SheetNumber/CopyNumber. SheetAbortA and SheetAbortB distinguish between the quality of the video which has been delivered. When the video is not guaranteed white, certain IOTs may elect to take protective measures. In either case, the image is retried. Continued failure will be referred to the operator by the PSP. If the operator elects to continue, a marker image will be delivered to the output in place of the failed image.

A *current* job, *next* job, or *interrupted* job may be aborted. A *previous* job, i.e., one which has completed imaging but is not yet fully delivered to the final destination, cannot be aborted. When a current job is aborted, all sheets in process after and including the specified SheetNumber/CopyNumber may be diverted to the scratch sheet destination. The IOT will continue with the next job. If no further jobs are programmed, the IOT will cycle down. JobAbortWithOutRecovery is specified for a total abort. If the job is current, both the imaged and unimaged portion of the job are to be abandoned. When JobAbortWithRecovery is specified, the IOT must monitor delivery to the final destination of the sheets already imaged, and attempt recovery if a jam is detected. Typically, JobAbortWithRecovery is specified when PspSheetBankAbort is being used to effect a set-boundary job interrupt, and that part of the job already imaged must be preserved to be recombined with the balance of the job to be reprogrammed and printed at a later time (refer to section 6.4.6). The IOT completes a job abort, with or without recovery, by deleting all banks of the aborted job. When AllJobsAbort is specified, all banks of all jobs are deleted, and the IOT cycles down.

PSPWRITEIOTMEMORY (0D)

PARAMETERS EndOfBlock (1) StartAddress (4) Length (1) Data (1 .. 117)

ORIGINATOR PSP

FUNCTION Allows the PSP to write to the IOT memory.

TRANSMISSION ORDER:

	ID CODE	EndOf Block	StartAddress				Length	Data		
Byte Nos.	B0	B1	B2	B3	B4	B5	B6	B7	- - -	B(7 + Length - 1)

DEFINITION:

PSPWRITEIOTMEMORY	::= IDCODE EndOfBlock StartAddress Length Data
IDCODE	::= B0 --byte 0
B0	::= 0D hex
EndOfBlock	::= B1 -- marks the last HDLC frame of this memory block transfer.
B1	::= No | Yes | Unused
No	::= 00 hex
Yes	::= 01 hex
Unused	::= [02 .. FF] hex
StartAddress	::= B2B3B4B5
B2B3B4B5	::= [00000000 .. FFFFFFFF] hex
Length	::= B6
B6	::= [01 .. 75] hex -- Number of data bytes in this command.(max. = 117 decimal)
Data	::= B7..B(7 + LengthMinusOne)
B7..	::= {[00 .. FF]1 .. [00 .. FF]$_{(1 + LengthMinusOne)}$} hex -- Variable number of bytes.

RESPONSE None (May be response to IotRequestMemory)

APPLICATION NOTES

Used by the PSP, in response to IotRequestMemory, to download. Multiple messages may be required to transfer the entire block of memory. StartAddress is incremented by the maximum block size each time a subsequent message, for a given block request, is sent by the PSP. Thus, StartAddress is always the memory address of the first data byte in the current message. The EndOfBlock field is used to indicate the last message in the sequence for a given block request.

PSPREQUESTIOTACTION (0E)

PARAMETERS Device (1) Set (1) Action (1) Data (2)

ORIGINATOR PSP

FUNCTION To request action of the IOT regarding certain devices or procedures.

TRANSMISSION ORDER:

	ID CODE	Device	Set	Action	Data	
Byte Nos.	B0	B1	B2	B3	B4	B5

DEFINITION:

PSPREQUESTIOTACTION	::=	IDCODE Device Set Action Data
IDCODE	::=	B0 --byte 0
B0	::=	0E hex
Device Set Action Data	::=	B1 B2 B3 B4B5
B1 B2 B3 B4B5	::=	LEDS LedSet LedAction LedMask
		| COUNTERS Counters CounterAction CounterData
		| FEEDERS Feeder FeederAction FeederData
		| DESTINATIONS Destination DestinationAction DestinationData
		| IOTDISPLAY DisplaySet DisplayAction DisplayData
		| CONTRASTCONTROL ContrastSet ContrastAction ProgrammableData
		| SCHEDULER Scheduler SchedulerAction SchedulerData
		| FINISHER FinisherSet FinisherAction FinisherData
		| ENERGYSAVER EnergySaverSet EnergySaverAction EnergySaverData
		| SPARE
LEDS	::=	01 hex -- B1
LedSet	::=	LedSet00 | LedSet01 | LedSet02 | Spare --B2
LedSet00	::=	00 hex
LedSet01	::=	01 hex
LedSet02	::=	02 hex
Spare	::=	[03 .. FF] hex
LedAction	::=	On | Off | Blink | Spare -- B3
On	::=	00 hex
Off	::=	01 hex
Blink	::=	02 hex

Spare	::= [03 .. FF] hex
LedMask	::= [0000000000000000 .. .. 1111111111111111]
COUNTERS	::= 02 hex -- B1
Counter	::= Counter00 | Counter01 | Counter02 | Spare -- B2
Counter00	::= 00 hex
Counter01	::= 01 hex
Counter02	::= 02 hex
Spare	::= [03 .. FF] hex
CounterAction	::= CountUp | CountDown | InitVal | Disable | AlphaVal | Spare -- B3
CountUp	::= 00 hex
CountDown	::= 01 hex
InitVal	::= 02 hex
Disable	::= 03 hex
AlphaVal:	:= 04 hex
Spare	::= [05 .. FF] hex
CounterData	::= [00 .. 9999] decimal -- B4B5
FEEDERS	::=03 hex
Feeder	::= Feeder0 | Feeder1 | Feeder2 | Feeder3 | Feeder4 | Feeder5 | Feeder6 | Feeder7
Feeder0	::= 00 hex
Feeder1	::= 01 hex
Feeder2	::= 02 hex
Feeder3	::= 03 hex
Feeder4	:: 04 hex
Feeder5	::= 05 hex
Feeder6	::= 06 hex
Feeder7	::= 07 hex
FeederAction	::= Raise | Lower | FeedFrom
Raise	::= 00 hex
Lower	::= 01 hex
FeedFrom	::= 02 hex -- start feeding from the designated feeder at the 1st opportunity.
FeederData	::= [0000 .. FFFF] hex -- Undefined
DESTINATIONS	::= [04] hex
Destination	::= Destination0 | Destination1 | Destination2 | Destination3 | Destination4 | Destination5 | Destination6 | Destination7
Destination0	::= 00 hex
Destination1	::= 01 hex
Destination2	::= 02 hex
Destination3	::= 03 hex
Destination4	::= 04 hex
Destination5	::= 05 hex
Destination6	::= 06 hex
Destination7	::= 07 hex
DestinationAction	::= Raise | Lower
Raise	::= 00 hex
Lower	::= 01 hex

DeliverTo	::=	02 hex -- start delivering to the designated destination at the 1st opportunity.
DestinationData	::=	[0000 .. FFFF] hex -- Undefined
IOTDISPLAY	::=	05 hex
DisplaySet	::=	00 hex
DisplayAction	::=	Clear | Display
Clear	::=	00 hex
Display	::=	01 hex
DisplayData	::=	MessageNumber
MessageNumber	::=	[0000 .. 0019] hex -- IOT specific.

-- The following message numbers are reserved for the use indicated. Exact text to be IOT specific.

0000 : Sample request granted

0001 : Sample request denied

0002 : See operator display for fault

0003 : See operator display for set-up instructions

CONTRASTCONTROL	::=	06 hex
ContrastSet	::=	00 hex -- not applicable
ContrastAction	::=	Normal | Darker | Lighter | Programmable | Spare
Normal	::=	00 hex
Darker	::=	01 hex
Lighter	::=	02 hex
Programmable	::=	03 hex -- see ProgrammableData
Spare	::=	[04 .. FF] hex
ProgrammableData	::=	ContrastSetting
ContrastSetting	::=	[0000 .. FFFF] hex -- definition of contrast range is IOT specific.
SCHEDULER	::=	07 hex
Scheduler	::=	00 hex -- not applicable
SchedulerAction	::=	MakeSample | Spare
MakeSample	::=	01 hex
Spare	::=	[02 .. FF] hex
SchedulerData	::=	0000 hex -- not applicable
FINISHER	::=	08 hex
FinisherSet	::=	Bindexer | Spare
Bindexer	::=	00 hex
Spare	::=	[01 .. FF] hex
FinisherAction	::=	PowerUp | PowerDown | SetTapeLength | Spare
PowerUp	::=	00 hex
PowerDown	::=	01 hex
SetTapeLength	::=	02 hex
Spare	::=	[03 .. FF] hex

FinisherData	::=	[0000 .. FF] hex - refer to "Application notes"
ENERGYSAVER	::=	09 hex
EnergySaverSet	::=	00 hex -- not applicable
EnergySaverAction	::=	RestoreFullPower | GoToNextLowerEnergyLevel | Spare
RestoreFullPower	::=	00 hex
GoToNextLowerEnergyLevel	::	01 hex
Spare	::=	[02 .. FF] hex
EnergySaverData	::=	0000 hex -- not applicable
SPARE	::=	[0A .. FF] hex

RESPONSE IOTOPERATIONALINFO InformationType DataField

APPLICATION NOTES

SCHEDULER: A MakeSample request received by the IOT when there are no active banks will be rejected. A MakeSample request received when there are active banks, will cause the IOT to make a sample at the first opportunity, and deliver it to the destination which is designated by the IOT to receive samples. There are two cases:

1. The IOT is in CycledUpPrinting state and TaskInProcess substate: the IOT will select an opportunity in the sheet sequence to make a sample and then continue with the normal image sequence.
2. The IOT is in CycledDownStandby state and TaskIncomplete substate: the IOT must cycle-up, make the sample, and then cycle down. To accomplish this, the PSP must follow the MakeSample request with PspRequestIotStateChange/CycleUp. When cycled-up the IOT Hints (refer to section 6.5) the sample image, the PSP responds with Print sample image, and, when appropriate, the IOT requests the video for the sample image. The PSP must send PspRequestIotStateChange/CycleDown to end the MakeSample operation. If any Hint messages occur for images following the sample image in the normal job sequence, the PSP must respond with Print deadcycle.

Note that under case 20, a sample copy from an incomplete job is being produced; however, the job does not resume after the sample is made. The operator may, however, first resume the job and then issue a MakeSample request, in which case the sample will be made as in case 1.

If the IOT is in CycledDownStandby and TaskComplete, a proof copy may be made by treating it as an ordinary job.

COUNTERS: The CounterActions are to interpreted as follows:

CountUp:	the IOT is to increment the designated counter.
CountDown:	the IOT is to decrement the designated counter.
InitVal:	the IOT is to initialize the designated counter to the value in the CounterData field.
Disable:	the IOT is to disable, i.e., not use, the designated counter.

AlphaVal: the IOT is to initialize the designated counter with the code of an alpha-numeric character contained in the CounterData field.

Note that CountUp, CountDown, and InitVal accommodate counters which are used exclusively for tallying. AlphaVal accommodates counters (registers) which are used exclusively to display characters on 7-segment displays. In the latter case, each message can display two characters, as represented by the two bytes of the CounterDataField. The character coding is to be per Ref. 5, section 6.2, International Reference Version. The lower order bit of the 7-bit character code is to be placed in bit d0 of the byte, and the upper order bit is to be placed in bit d6. Bit d7 is to be 0. The characters used are to be limited to those suitable for 7-segment displays.

FINISHER: FinisherAction is PowerUp or PowerDown, FinisherData is a power down time-out in IOT specific units. Application of the time-out is also IOT specific. For example, if received with a PowerUp command, the time-out could be started when the finisher next receives a job that does not specify binding; if received with a PowerDown command, it could be started upon completion of a binding job in progress, or immediately if a binding job is not in progress. Care must be taken in assigning meaning to the default data, 0000 hex, since this value will always be sent with a PowerUp command if no other value is specified.

When FinisherAction is SetTapeLength, FinisherData is tape length in IOT specific units.

PSPREQUESTIOTSTATECHANGE (0F)

PARAMETERS ChangeRequested (1)

ORIGINATOR PSP

FUNCTION To command the IOT to change machine state.

TRANSMISSION ORDER:

	ID CODE	Change Requested
Byte Nos.	B0	B1

DEFINITION:

PSPREQUESTIOTSTATECHANGE	::=	IDCODE ChangeRequested
IDCODE	::=	B0 --byte 0
B0	::=	0F hex
ChangeRequested	::	B1
B1	::=	[d7 .. d0]
d0	::=	CycleUp | CycleDown
CycleDown	::=	0 binary
CycleUp	::=	1 binary
d7 d6 d5 d4 d3 d2 d1	::=	Spare

RESPONSE IOTSTATEINFO MachineState Substates

APPLICATION NOTES Afterr request for machine state change, the IOT may optionally report the transient states CyclingUp or CyclingDown, but will always report the resulting final steady state (refer to IotStateInfo and section 6.4.1, IOT States).

Figure 6-1. **Example of PSPCONFIGURATION command in information field of HDLC frame**

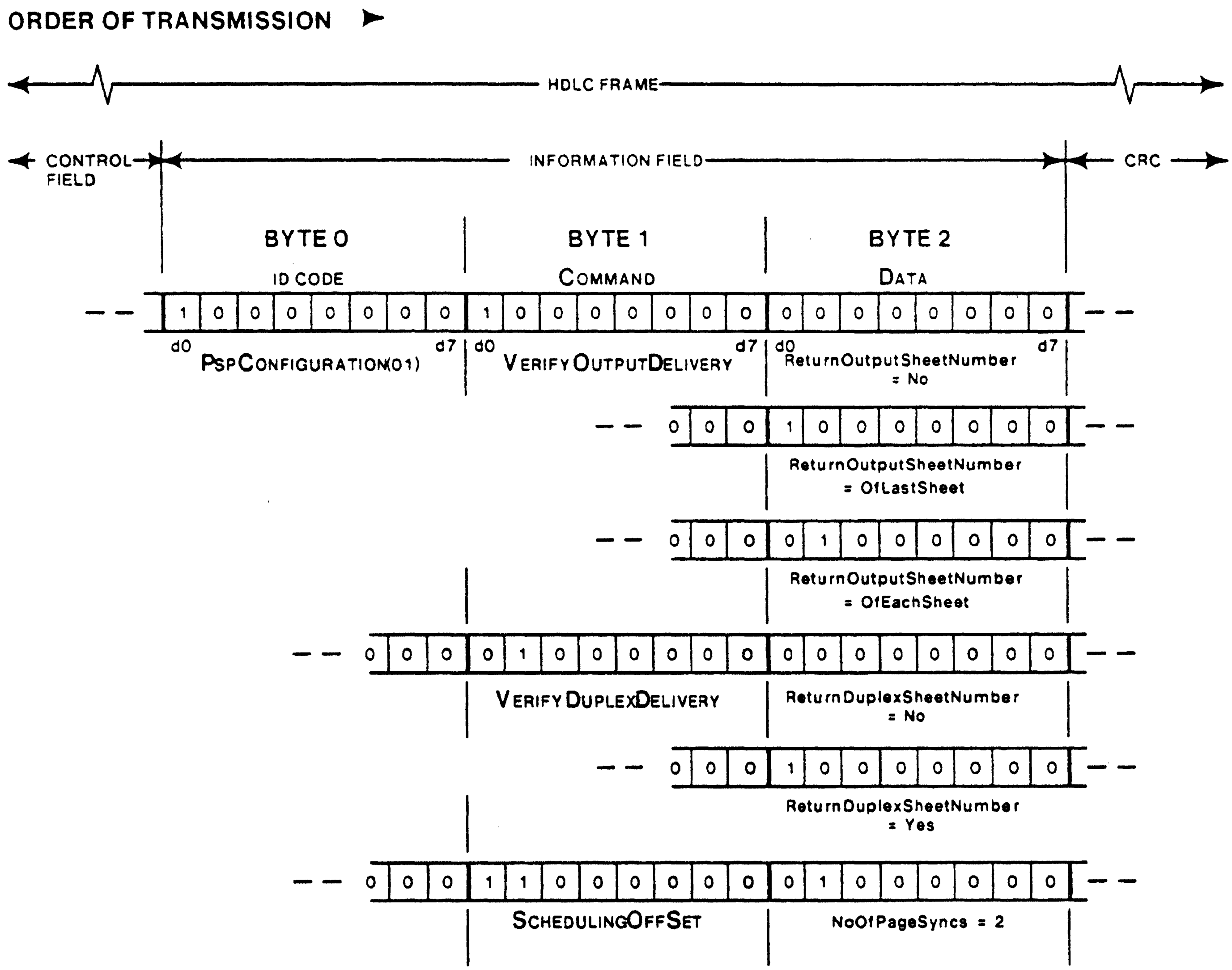
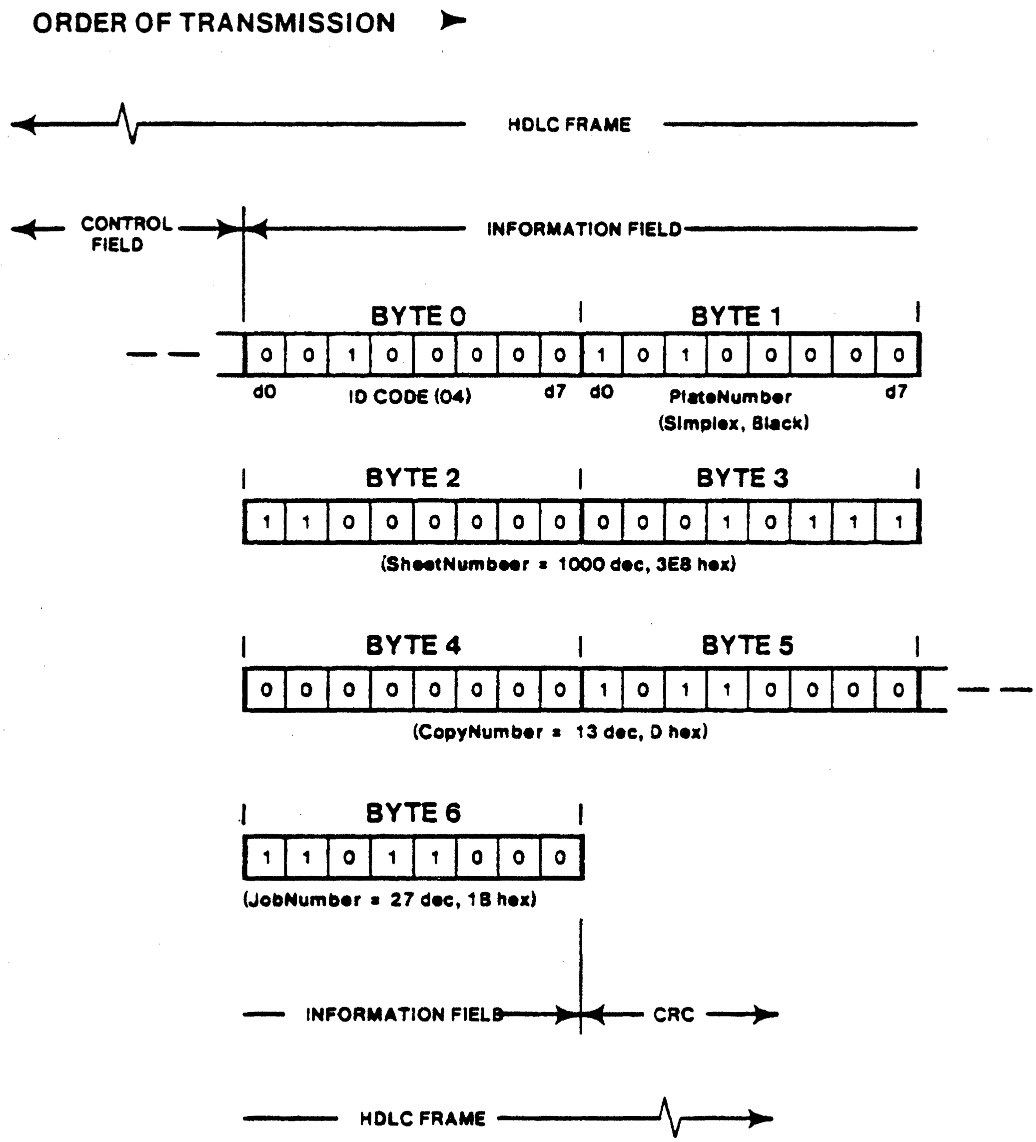


Figure 6-2. **Example of PSPPRINT command in information field of HDLC frame**



6.3.3 IOT Status messages

IOTCONFIGURATION (81)

PARAMETERS InformationType (1) DataField (4, 9, 10 or 16)

ORIGINATOR IOT

FUNCTION To provide the PSP with IOT configuration data.

TRANSMISSION ORDER:

	ID Code	Information Type	DataField					DataField		DataField		DataField
Byte Nos.	B0	B1	B2	B3	B4	B5	— —	B10	— —	B11	— —	B17

DEFINITION:

IOTCONFIGURATION	::= IDCODE InformationType DataField
IDCODE	::= B0 -- byte 0
B0	::= 81 hex
InformationType DataWord	::= B1 [B2 .. B16] --4 different formats
B1 [B2 .. B16]	::= CONFIGURATION ConfigInfo
	| MEDIAMATRIX MatrixInfo
	| DESTINATION0 DestinationInfo
	| DESTINATION1 DestinationInfo
	| DESTINATION2 DestinationInfo
	| DESTINATION3 DestinationInfo
	| DESTINATION4 DestinationInfo
	| DESTINATION5 DestinationInfo
	| DESTINATION6 DestinationInfo
	| DESTINATION7 DestinationInfo
	| FEEDER0 FeederInfo
	| FEEDER1 FeederInfo
	| FEEDER2 FeederInfo
	| FEEDER3 FeederInfo
	| FEEDER4 FeederInfo
	| FEEDER5 FeederInfo
	| FEEDER6 FeederInfo
	| FEEDER7 FeederInfo
	| SPARE SpareInfo
CONFIGURATION	::= 00 hex -- 17 byte format.
ConfigInfo	::= ConfigID DataLinkAddress DataLinkAckTime PaperPathLength SIFlength PowerDownWarning Resolution InterruptedJobs TotalJobs DuplexType ConfigInfoA BeltSpeed
ConfigID	::= B2
B2	::= [00 .. FF] hex
DataLinkAddress	::= B3
B3	::= [00 .. FF] hex
DataLinkAckTime	::= B4
B4	::= [00 .. FF] hex -- maximum, in milliseconds

```
PaperPathLength       ::= B5
    B5                ::= [00 .. FF] hex -- longest path from
                                            lotVideoHint to
                                            lotSheetDelivered, in number of
                                            page-times.
SIFlength             ::= B6 B7
    B6 B7             ::= [0000 .. FFFF] hex --Fast scan dimension of
                                               Standard Image Frame, in
                                               number of pixels.
PowerDownWarning      ::= B8 B9
    B8 B9             ::= [0000 .. FFFF] hex --minimum, in milliseconds.
Resolution            ::= B10 B11
    B10 B11           ::= [0000 .. FFFF] hex --in pixels/inch, isomorphic.
InterruptedJobs       ::= B12
    B12               ::= [00 .. FF] hex  -- max. number supported.
TotalJobs             ::= B13
    B13               ::= [00 .. FF] hex  -- max. no. of all jobs
                                             supported.
DuplexType            ::= B14
    B14               ::= SimplexOnly | RaceTrack | Storing
        SimplexOnly   ::= 00 hex
        RaceTrack     ::= 01 hex
        Storing       ::= 02 hex
ConfigInfoA           ::= B15
    B15               ::= d7d6d5d4d3d21d0
        d0            ::= FeedStyle
            FeedStyle ::= CutSheet | WebFeed
             CutSheet ::= 0 binary
             WebFeed  ::= 1 binary
        d3d2d1        ::= PaperRegMode
          PaperRegMode ::= Undefined | A | B | C | D | E | F -- Refer to
                                                         para 3.1.1.
             Undefined ::= 000 binary
             A         ::= 001 binary
             B         ::= 010 binary
             C         ::= 011 binary
             D         ::= 100 binary
             E         ::= 101 binary
             F         ::= 110 binary
        d5d4          ::= Spare
        d6            ::= VideoInterface
          VideoInterface ::= Parallel | Serial
             Parallel  ::= 0 binary -- implies byte clock at interface.
             Serial    ::= 1 binary -- implies pixel clock at interface.
        d7            ::= Spare
BeltSpeed             ::= B16B17
    B16B17            ::= [0000 .. FFFF] hex -- in mm per second
MEDIAMATRIX           ::= 01 hex     -- 10 byte format.
MatrixInfo            ::= PaperWidth SIFwidth PageTime
                              SchedulingOffset DuplexOffset Spare
    PaperWidth        ::= B2 B3 -- slow scan dimension, in millimeters.
        B2 B3         ::= [0000 .. FFFF] hex
    SIFwidth          ::= B4 B5
        B4 B5         ::= [0000 .. FFFF] hex  -- slow scan dimension of
                                                 Standard Image Frame, in
                                                 number of scan lines.
PageTime              ::= B6 B7
    B6 B7             ::= [0000 .. FFFF] hex  -- in milliseconds.
SchedulingOffset      ::= B8
```

B8	::=	[00 .. FF] hex -- The number of page sync positive transitions between PspPrint and IotVideoRequest as required by the IOT.
DuplexOffset	::=	B9
B9	::=	[00 .. FF] hex -- The number of page-times between IotVideoHint-Simplex and IotVideoHint-Duplex, as required by the IOT.
EndOfMatrix	::=	B10
B10	::=	No | Yes
No	::=	00 hex
Yes	::=	01 hex
DESTINATION0	::=	02 hex -- 5 byte format
DESTINATION1	::=	03 hex
DESTINATION2	::=	04 hex
DESTINATION3	::=	05 hex
DESTINATION4	::=	06 hex
DESTINATION5	::=	07 hex
DESTINATION6	::=	08 hex
DESTINATION7	::=	09 hex
DestinationInfo	::=	DeviceID Parameter ParameterData
DeviceID	::=	B2
B2	::=	NotImplemented | TopTray | Stacker MultibinSorter | Bindexer | Spare
NotImplemented	::=	00 hex
Parameter	::=	B3
B3	::=	00 hex -- not applicable
ParameterData	::=	B4B5
B4B5	::=	0000 hex -- not applicable
TopTray	::=	01 hex
Parameter	::=	B3
B3	::=	Unassigned | MaxPaperWidth | MaxPaperLength | MinPaperWidth | MinPaperLength | TopTrayAttributes | MaxCapacity | Spare
Unassigned	::=	00 hex
MaxPaperWidth	::=	01 hex
WidthParameter	::=	B4B5
B4B5	::=	Undefined | WidthDimension
Undefined	::=	0000 hex6
WidthDimension	::=	[0001 .. FFFF] hex -- Max width in mm.
MaxPaperLength	::=	02 hex
LengthParameter	::=	B4B5
B4B5	::=	Undefined | LengthDimension
Undefined	::=	0000 hex
LengthDimension	::=	[0001 .. FFFF] hex -- Max length in mm.
MinPaperWidth	::=	03 hex
WidthParameter	::=	B4B5
B4B5	::=	Undefined | WidthDimension
Undefined	::=	0000 hex

```
          WidthDimension   ::= [0001 .. FFFF]  hex  -- Min width in mm.
      MinPaperLength       ::= 04 hex
        LengthParameter    ::= B4B5
          B4B5             ::= Undefined | LengthDimension
            Undefined      ::= 0000 hex
            LengthDimension ::= [0001 .. FFFF]  hex -- Min length in mm.
      TopTrayAttributes    ::= 05 hex
        ParameterData      ::= B4B5
        B4B5               ::= Sequence  Offset  Spare
          Sequence         ::= d1d0
            d1d0           ::= Unassigned | 1-to-N | N-to-1 | Both
              Unassigned   ::= 00 binary
              1-to-N       ::= 01 binary
              N-to-1       ::= 10 binary
              Both         ::= 11 binary
            Offset         ::= d2
              d2           ::= No | Yes
                No         ::= 0 binary
                Yes        ::= 1 binary
            Spare          ::= [d15 .. d3]
      MaxCapacity          ::= 06 hex
        NumberOfSheets     ::= B4B5
          B4B5             ::= [0000 .. FFFF] hex
        Spare              ::= [07 .. FF] hex
      Stacker              ::= 02 hex
        Parameter          ::= B3
          B3               ::= Unassigned  | MaxPaperWidth
                               | MaxPaperLength | MinPaperWidth
                               | MinPaperLength | StackerAttributes
                               | MaxCapacity | Spare
          Unassigned       ::= 00 hex
          MaxPaperWidth    ::= 01 hex
            WidthParameter ::= B4B5
              B4B5         ::= Undefined | WidthDimension
              Undefined    ::= 0000 hex
              WidthDimension ::= [0001 .. FFFF]  hex  -- Max width in mm.
    MaxPaperLength         ::= 02 hex
      LengthParameter      ::= B4B5
          B4B5             ::= Undefined | LengthDimension
            Undefined      ::= 0000 hex
            LengthDimension ::= [0001 .. FFFF]  hex -- Max length in mm.
    MinPaperWidth          ::= 03 hex
      WidthParameter       ::= B4B5
```

```
            B4B5                    ::= Undefined | WidthDimension
                Undefined           ::= 0000 hex
                WidthDimension      ::= [0001 .. FFFF]  hex  -- Min width in mm.
        MinPaperLength              ::= 04 hex
            LengthParameter         ::= B4B5
                B4B5                ::= Undefined | LengthDimension
                    Undefined       ::= 0000 hex
                    LengthDimension ::= [0001 .. FFFF]  hex -- Min length in mm.
        StackerAttributes           ::= 05 hex
            ParameterData           ::= B4B5
                B4B5                ::= Sequence  Offset  Spare
                    Sequence        ::= d1d0
                      d1d0          ::= Unassigned | 1-to-N | N-to-1 | Both
                        Unassigned  ::= 00 binary
                        1-to-N      ::= 01 binary
                        N-to-1      ::= 10 binary
                        Both        ::= 11 binary
            Offset                  ::= d2
                d2                  ::= No | Yes
                    No              ::= 0 binary
                    Yes             ::= 1 binary
            Spare                   ::= [d15 .. d3]
        MaxCapacity                 ::= 06 hex
            NumberOfSheets          ::= B4B5
                B4B5                ::= [0000 .. FFFF] hex
        Spare                       ::= [07 .. FF] hex
MultibinSorter                      ::= 03 hex
    Parameter                       ::= B3
        B3                          ::= Unassigned  | MaxPaperWidth
                                        | MaxPaperLength | MinPaperWidth
                                        | MinPaperLength | SorterAttributes
                                        | BinComplement | CapacityPerBin
                                        | Spare
        Unassigned                  ::= 00 hex
        MaxPaperWidth               ::= 01 hex
            WidthParameter          ::= B4B5
                B4B5                ::= Undefined | WidthDimension
                    Undefined       ::= 0000 hex
                    WidthDimension  ::= [0001 .. FFFF]  hex  -- Max width in mm.
```

```
        MaxPaperLength          ::= 02 hex
            LengthParameter     ::= B4B5
              B4B5              ::= Undefined | LengthDimension
                Undefined       ::= 0000 hex
                LengthDimension ::= [0001 .. FFFF] hex -- Max length in mm.
        MinPaperWidth           ::= 03 hex
            WidthParameter      ::= B4B5
              B4B5              ::= Undefined | WidthDimension
                Undefined       ::= 0000 hex
                WidthDimension  ::= [0001 .. FFFF] hex -- Min width in mm.
        MinPaperLength          ::= 04 hex
            LengthParameter     ::= B4B5
              B4B5              ::= Undefined | LengthDimension
                Undefined       ::= 0000 hex
                LengthDimension ::= [0001 .. FFFF] hex -- Min length in mm.
        SorterAttributes        ::= 05 hex
              B4B5              ::= Sequence Offset Spare
                Sequence        ::= d1d0
                  d1d0          ::= Unassigned | 1-to-N | N-to-1 | Both
                    Unassigned  ::= 00 binary
                    1-to-N      ::= 01 binary
                    N-to-1      ::= 10 binary
                    Both        ::= 11 binary
                Offset          ::= d2
                  d2            ::= No | Yes
                    No          ::= 0 binary
                    Yes         ::= 1 binary
              Spare             ::= [d15 .. d3]
        BinComplement           ::= 06 hex
            NumberOfBins        ::= B4B5
              B4B5              ::= [0000 .. FFFF] hex
        CapacityPerBin          ::= 07 hex
            NumberOfSheets      ::= B4B5 -- maximum
              B4B5              ::= [0000 .. FFFF] hex
        Spare                   ::= [08 .. FF] hex
    Bindexer                    ::= 04 hex
        Parameter               ::= B3
            B3                  ::= Unassigned | MaxPaperWidth
                                    | MaxPaperLength | MinPaperWidth
                                    | MinPaperLength | BindexerAttributes
                                    | BinComplement | CapacityPerBin
                                    | Spare
```

```
Unassigned              ::= 00 hex
MaxPaperWidth           ::= 01 hex
   WidthParameter       ::= B4B5
      B4B5              ::= Undefined | WidthDimension
        Undefined       ::= 0000 hex
        WidthDimension  ::= [0001 .. FFFF]  hex  -- Max width in mm.
MaxPaperLength          ::= 02 hex
   LengthParameter      ::= B4B5
      B4B5              ::= Undefined | LengthDimension
        Undefined       ::= 0000 hex
        LengthDimension ::= [0001 .. FFFF]  hex -- Max length in mm.
MinPaperWidth           ::= 03 hex
   WidthParameter       ::= B4B5
      B4B5              ::= Undefined | WidthDimension
        Undefined       ::= 0000 hex
        WidthDimension  ::= [0001 .. FFFF]  hex  -- Min width in mm.
MinPaperLength          ::= 04 hex 6
   LengthParameter      ::= B4B5
      B4B5              ::= Undefined | LengthDimension
        Undefined       ::= 0000 hex
        LengthDimension ::= [0001 .. FFFF]  hex -- Min length in mm.
BindexerAttributes      ::= 05 hex
   B4B5                 ::= Sequence  Offset  StitchingOptions
                            BindingOptions  Spare
      Sequence          ::= d1d06
        d1d0            ::= Unassigned | 1-to-N | N-to-1 | Both
          Unassigned    ::= 00 binary
          1-to-N        ::= 01 binary
          N-to-1        ::= 10 binary
          Both          ::= 11 binary
   Offset               ::= d2
      d2                ::= No | Yes
        No              ::= 0 binary
        Yes             ::= 1 binary
   StitchingOptions     ::= d4d3
      d4d3              ::= None | PortraitSingleStitch
                            | LandscapeSingleStitch | DoubleStitch
        None            ::= 00 binary
        PortraitSingleStitch::= 01 binary
        LandscapeSingle
        Stitch          ::= 10 binary
        DoubleStitch    ::= 11 binary
```

```
        BindingOptions          ::= d6d5
        d6d5                    ::= None | BindOnly | DoubleStitchAndBind
                                    | Spare
         None                   ::= 00 binary
         BindOnly               ::= 01 binary
         DoubleStitchAndBind    ::= 10 binary
         Spare                  ::= 11 binary
        Spare                   ::= [d15 .. d7]
    BinComplement               ::= 06 hex
        NumberOfBins            ::= B4B5
          B4B5                  ::= [0000 .. FFFF] hex
    CapacityPerBin              ::= 07 hex
        NumberOfSheets          ::= B4B5  -- maximum
          B4B5                  ::= [0000 .. FFFF] hex
    Spare                       ::= [08 .. FF] hex
Spare                           ::= [05 .. FF] hex    -- B2
FEEDER0                         ::= 0A hex     -- 11 byte format
FEEDER1                         ::= 0B hex
FEEDER2                         ::= 0C hex
FEEDER3                         ::= 0D hex
FEEDER4                         ::= 0E hex
FEEDER5                         ::= 0F hex
FEEDER6                         ::= 10 hex
FEEDER7                         ::= 11 hex6
    FeederInfo                  ::= DeviceID   MaxPaperWidth
                                    MaxPaperLength   MinPaperWidth
                                    MinPaperLength  FeederAttributes
        DeviceID                ::= B2
          B2                    ::= NotImplemented | Implemented
            NotImplemented      ::= 00 hex
            Implemented         ::= 01 hex
    FeederAttributes            ::= B3
        B3                      ::= Transparencies  DrilledPaper  Spare
          Transparencies        ::= d0
            d0                  ::= No | Yes
              No                ::= 0 binary
              Yes               ::= 1 binary
        DrilledPaper            ::= d1
          d1                    ::= No | Yes
            No                  ::= 0 binary
            Yes                 ::= 1 binary
```

Spare	::=	[d7 .. d2]
MaxPaperWidth	::=	B4B5
B4B5	::=	[0000 .. FFFF] hex -- Max width in mm.
MaxPaperLength	::=	B6B7
B6B7	::=	[0000 .. FFFF] hex -- Max length in mm.
MinPaperWidth	::=	B8B9
B8B9	::=	[0000 .. FFFF] hex -- Min width in mm.
MinPaperLength	::=	B10B11
B10B11	::=	[0000 .. FFFF] hex -- Min length in mm.
Spare	::=	[12 .. FF] hex -- B1

IN RESPONSE TO PSPCONFIGURATION Volunteered

APPLICATION NOTES

The formats of this message, corresponding to the four different InformationTypes, are as follows:

CONFIGURATION requires 17 bytes (excluding ID Code):

Byte Nos.	B0	B1	B2	B3	B4	B5	B6–B7
	ID Code	Configur-ation	ConfigID	DataLink Address	DataLink AckTime	PaperPath Length	SIFlength

Byte Nos.	B8–B9	B10–B11	B12	B13	B14	B15	B16–B17
	PowerDownWarning	Resolution	Interrupted Jobs	Total Jobs	DuplexType	ConfigInfo	BeltSpeed

All IOTs must use the default *DataLinkAddress*, 01 hex, until otherwise noted herein.

DataLinkAckTime is the time which the IOT guarantees not to exceed when acknowledging (positively or negatively) a received data link frame which requires acknowledgement. This becomes the acknowledgement time-out employed by the PSP data link transmitting function (refer to section 5.3.9-e).

PowerDownWarning is the minimum warning time which the IOT desires in advance of a scheduled power-down by the PSP.

InterruptedJobs specifies the maximum number of job interrupts which the IOT can manage concurrently. *TotalJobs* specifies the maximum number of jobs of all categories combined which the IOT can manage concurrently. Refer to sections 6.4.3 and 6.4.6 for definition of job types and job interrupt protocol.

DuplexType under CONFIGURATION is to be used in conjunction with *DuplexOffset* under MEDIAMATRIX. If DuplexType is *RaceTrack*, the PSP will use the figure stated in DuplexOffset as a fixed parameter. If DuplexType is *Storing*, the IOT must state the minimum offset. The PSP will use this figure as the minimum offset and also will infer from it the intermediate offsets to be used while the duplex paper path is being filled, as well as the maximum offset to be used when the duplex paper path is filled.

The leading edge and trailing edge aspects of paper registration in *PaperRegMode* under configInfoA may be regarded as not applicable to web feed IOT's.

MEDIAMATRIX requires 10 bytes (excluding ID Code):

Byte Nos.	ID CODE	Media Matrix	PaperWidth		SIFwidth	
	B0	B1	B2	B3	B4	B5

Byte Nos.	PageTime		Sched Offset	Duplex Offset	End of Matrix
	B6	B7	B8	B9	B10

MEDIAMATRIX is the vehicle by which the PSP knows which Standard Image Frame (and related parameters) to expect when it specifies the paper dimensions in PspNextBankRequest. An IOT may implement more than one Standard Image Frame. The procedure for changing from one to another is IOT specific, and may or may not require cycling down. It is expected that many IOTs will implement only a single Standard Image Frame. Each paper width requires a separate message. The number of messages is, thus, generically indeterminate, so an EndOfMatrix field is included. An IOT must designate a Standard Image Frame width of zero to indicate that it is a gapless IOT and will generate **page sync** to match the paper width requested by the PSP via PspNextBankRequest.

For web feed IOTs, *PaperWidth* is to be interpreted as web width. For simultaneous duplex IOTs, *DuplexOffset* is to be interpreted as the offset between imaging stations.

DESTINATION requires 5 bytes (excluding ID Code):

Byte Nos.	ID CODE	Destination Number	DeviceID	Parameter Number	ParameterData	
	B0	B1	B2	B3	B4	B5

Each destination parameter requires a separate message, and each destination requires a separate set of messages. Up to seven parameters per destination are prescribed herein (e.g., bindexer). The format provides for future expansion to 255 parameters per destination.

FEEDER requires 11 bytes (excluding ID Code) and a separate message for each feeder in order to deliver all of the information:

Byte Nos.	ID CODE	Feeder Number	DeviceID	Feeder Attributes	MaxPaperWidth		MaxPaperLength	
	B0	B1	B2	B3	B4	B5	B6	B7

Byte Nos.	MinPaperWidth		MinPaperLength	
	B8	B9	B10	B11

IOTMETARESET (82)

PARAMETERS IotStatus (2) ResetMode (1)

ORIGINATOR IOT

FUNCTION To inform the PSP of the specific unrecoverable error detected in the IOT and that an IOT reset is necessary. The IOT either requests a hardwired reset or indicates that it will reset itself.

TRANSMISSION ORDER:

	ID CODE	IotStatus CourseCode	IotStatus Finecode	ResetMode
Byte Nos.	B0	B1	B2	B3

DEFINITION:

IOTMETARESET	::=	IDCODE IotStatus ResetMode
IDCODE	::=	B0 --byte 0
B0	::=	82 hex
IotStatus	::=	CourseCode FineCode
CourseCode	::=	B1
CourseCode	::=	[00 .. FF] hex -- IOT specific
FineCode	::=	B2
FineCode	::=	[00 .. FF] hex -- IOT specific
ResetMode	::=	B3
B3	::=	SelfReset | RequestPspReset
SelfReset	::=	00 hex
RequestPspReset	::=	01 hex

IN RESPONSE TO Volunteered

APPLICATION NOTES

When RequestPspReset is specified in the ResetMode byte, the PSP is to perform a hardwired reset of the IOT by means of the Power Control cirduit in the physical interface.

This message may be sent in the Un-numbered Information format of the HDLC frame to preclude the possibility of invoking the Data Link time-out and retransmission strategy during implied exception conditions.

IOTVIDEOHINT (83)

PARAMETERS PlateNumber (1) SheetNumber (2) CopyNumber (2) JobNumber (1)

ORIGINATOR IOT

FUNCTION To hint the video information required for a subsequent page time of the IOT.

TRANSMISSION ORDER:

	ID CODE	Plate Number	SheetNumber		CopyNumber		Job Number
Byte Nos.	B0	B1	B2	B3	B4	B5	B6

DEFINITION:

IOTVIDEOHINT	::= IDCODE PlateNumber SheetNumber CopyNumber JobNumber
IDCODE	::= B0 --byte 0
B0	::= 83 hex
PlateNumber	::= B1
B1	::= PageMode PlateColor PlateResolution
PageMode	::= d1 d0
d1 d0	::= Duplex | Simplex | SimultaneousDuplex | Spare1
Duplex	::= 00 binary
Simplex	::= 01 binary
SimultaneousDuplex	::= 10 binary
Spare1	::= 11 binary
PlateColor	::= d5 d4 d3d2
PlateDeadCycle	::= 000000 binary
d2	::= NotColor0 | Color0
NotColor0	::= 0 binary
Color0	::= 1 binary
d3	::= NotColor1 | Color1
NotColor1	::= 0 binary
Color1	::= 1 binary
d4	::= NotColor2 | Color2
NotColor2	::= 0 binary
Color2	::= 1 binary
d5	::= NotColor3 | Color3
NotColor3	::= 0 binary
Color3	::= 1 binary

ResolutionMode	:: =	d7 d6
d6	:: =	SimplexRes1 | SimplexRes2
SimplexRes1	:: =	0 binary -- IOT specific
SimplexRes2	:: =	1 binary -- IOT specific
d7	:: =	DuplexRes1 | DuplexRes2
DuplexRes1	:: =	0 binary -- IOT specific
DuplexRes2	:: =	1 binary -- IOT specific
SheetNumber	:: =	B2 B3 -- Sheet to be committed to the paper path by the IOT upon receipt of PspPrint Command specifying same Sheet, Plate, and Copy Number.
B2 B3	:: =	SheetDeadcycle | SheetIdentifier
SheetDeadCycle	:: =	0000 hex
SheetIdentifier	:: =	[0001 .. FFFF]
CopyNumber	:: =	B4B5 -- See note under SheetNumber
B4B5	:: =	SampleCopy |CopyIdentifier
SampleCopy	:: =	0000 hex
CopyIdentifier	:: =	[0001 .. FFFF] hex
JobNumber	:: =	B6
B6	:: =	Undefined | JobIdentifier
Undefined	:: =	00 hex
JobIdentifier	:: =	[01 .. FF] hex

IN RESPONSE TO Volunteered

APPLICATION NOTES

Refer to sections 6.4 and 6.5 for timing requirements, numbering and sequencing conventions, and examples of this message in operating sequences.

The convention for assigning color designations under PlateNumber/PlateColor is as follows: For single color printing, usually black, Color0 shall designate the single color. For single hi-lite color printing, Color0 shall designate the principal color, usually black; and Color1 shall designate the hi-lite color (the hue of the hi-lite color is IOT specific). For multicolor hi-lite color printing, Color0 shall designate the principal color, usually black; and Color1 through Color3 shall designate IOT specific hi-lite colors. For full color printing, Color0 through Color3 are black, magenta, cyan, and yellow, respectively.

IOTVIDEOREQUEST (84)

PARAMETERS PlateNumber (1) SheetNumber (2) CopyNumber (2) JobNumber (1)

ORIGINATOR IOT

FUNCTION To request video data for the next page sync signal. The Sheet, Plate, and Copy number sent bythe IOT may be out ofthe normal expected order if sent to different output destinations. Under exception conditions, the Sheet, Plate, Copy, and Job number sent by the IOT may be repeated in subsequent page-times.

TRANSMISSION ORDER:

	ID CODE	Plate Number	SheetNumber		CopyNumber		Job Number
Byte Nos.	B0	B1	B2	B3	B4	B5	B6

DEFINITION:

IOTVIDEOREQUEST	::= IDCODE PlateNumber SheetNumber CopyNumber JobNumber
IDCODE	::= B0 -- byte 0
B0	::= 84 hex
PlateNumber	::= B1
B1	::= PageMode PlateColor PlateResolution
PageMode	::= d1 d0
d1 d0	::= Simplex | Duplex | SimultaneousDuplex | Spare1
Duplex	::= 00 binary
Simplex	::= 01 binary
SimultaneousDuplex	::= 10 binary
Spare1	::= 11 binary
PlateColor	::= d5 d4 d3d2
PlateDeadCycle	::= 000000 binary
d2	::= NotColor0 | Color0
NotColor0	::= 0 binary
Color0	::= 1 binary
d3	::= NotColor1 | Color1
NotColor1	::= 0 binary
Color1	::= 1 binary
d4	::= NotColor2 | Color2
NotColor2	::= 0 binary
Color2	::= 1 binary
d5	::= NotColor3 | Color3
NotColor3	::= 0 binary
Color3	::= 1 binary
ResolutionMode	::= d7 d6
d6	::= SimplexRes1 | SimplexRes2
SimplexRes1	::= 0 binary -- IOT specific
SimplexRes2	::= 1 binary -- IOT specific
d7	::= DuplexRes1 | DuplexRes2
DuplexRes1	::= 0 binary -- IOT specific
DuplexRes2	::= 1 binary -- IOT specific

SheetNumber	::=	B2 B3
B2 B3	::=	SheetDeadcycle | SheetIdentifier
SheetDeadCycle	::=	0000 hex
SheetIdentifier	::=	[0001 .. FFFF] hex
CopyNumber	::=	B4B5
B4B5	::=	SampleCopy | CopyIdentifier
SampleCopy	::=	0000 hex
CopyIdentifier	::=	[0001 .. FFF] hex
JobNumber	::=	B6
B6	::=	Undefined | JobIdentifier
Undefined	::=	00 hex
JobIdentifier	::=	[01 .. FF] hex

IN RESPONSE TO PSPPRINT PlateNumber SheetNumber CopyNumber JobNumber

APPLICATION NOTES

Refer to sections 6.4 and 6.5 for timing requirements, numbering and sequencing conventions, and examples of this message in operating sequences.

The convention for assigning color designations under *PlateNumber/PlateColor* is as follows: For single color printing, usually black, Color0 shall designate the single color. For single hi-lite color printing, Color0 shall designate the principal Color, usually black; and Color1 shall designate the hi-lite color (the hue of the hi-lite color is IOT specific). For multicolor hi-lite color printing, Color0 shall designate the principal color, usually black, and Color1 through Color3 shall designate IOT specific hi-lite colors. For full color printing, Color0 through Color3 are black, magenta, cyan, and yellow, respectively.

IOTWRITEMEMORYTOPSP (86)

PARAMETERS EndOfBlock (1) StartAddress (4) Length (1) Data (1 .. 117)

ORIGINATOR IOT

FUNCTION To transmit contents of IOT Memory to the PSP either voluntarily or as requested.

TRANSMISSION ORDER:

	ID CODE	EndOf Block	StartAddress				Length	Data - - -				
Byte Nos.	B0	B1	B2	B3	B4	B5	B6	B7	-	-	-	B(7 + Length - 1)

DEFINITION:

IOTMEMORYREADRESPONSE	::= IDCODE EndOfBlock StartAddress Length Data
IDCODE	::= B0 --byte 0
B0	::= 0D hex
EndOfBlock	::= B1 -- marks the last HDLC frame of this memory block transfer.
B1	::= No | Yes | Unused
No	::= 00 hex
Yes	::= 01 hex
Unused	::= [02 .. FF] hex
StartAddress	::= B2B3B4B5
B2B3B4B5	::= [00000000 .. FFFFFFFF] hex
Length	::= B6
B6	::= [00 .. 75] hex -- Number of data bytes in this message, (max. = 117 decimal)
Data	::= B7..B(7 + LengthMinusOne)
B7..	::= {[00 .. FF]1 .. [00 .. FF]$_{(1 + LengthMinusOne)}$} hex -- Variable number of bytes.

IN RESPONSE TO PSPREADIOTMEMORY StartAddress EndAddress or Volunteered

APPLICATION NOTES

Multiple messages may be required to transfer the entire block of memory. The EndOfBlock field is used to indicate the last message in the sequence.

IotStateInfo (87)

PARAMETERS MachineState (1) Substates (1)

ORIGINATOR IOT

FUNCTION To volunteer information following a power up, a major or minor IOT state change, detection of a fault, or in response to a PspReadIotState or PspRequestIotStateChange.

TRANSMISSION ORDER:

	ID CODE	Machine States	Substates
Byte Nos.	B0	B1	B2

DEFINITION:

IotStateInfo	::= IdCode MachineState Substates
IdCode	::= B0 --byte 0
B0	::= 87 hex
MachineState Substates	::= B1 B2
B1 B2	::= CycledDownStandby Substates | CycledDownNotReady Substates | CyclingDown Substates | CyclingUp Substates | CycledUpPrinting Substates | CycledUpNotReady Substates
CycledDownStandby	::= 00 hex
CycledDownNotReady	::= 01 hex
CyclingDown	::= 02 hex
CyclingUp	::= 03 hex
CycledUpPrinting	::= 04 hex
CycledUpNotReady	::= 05 hex
Spare	::= [06 -- FF] hex
Substates	::= d7 .. d0
d1d0	::= JobStates
TaskStates	::= TaskComplete | TaskInProcess | TaskReadyForRestart | TaskIncomplete
TaskComplete	::= 00 binary
TaskInProcess	::= 01 binary
TaskReadyForRestart	::= 10 binary
TaskIncomplete	::= 11 binary
d2	::= FaultStates
FaultStates	::= FaultNotDetected | FaultDetected
FaultNotDetected	::= 0 binary
FaultDetected	::= 1 binary
d3	::= ProductivityStates
ProductivityStates	::= Productive | NonProductive

Productive	::=	0 binary
NonProductive	::=	1 binary
d7d6d5d4	::=	Spare

IN RESPONSE TO PSPREADIOTSTATE, PSPREQUESTIOTSTATECHANGE, or Volunteered

APPLICATION NOTES

This message contains the machine states and substates of an IOT which must be observable at the generic interface. An IOT may have more internal states, which need not be reported at the interface, but they must fall within the definitions given in the following section (also refer to section 6.4.1, IOT States).

The substates are defined as follows:

TaskComplete All sheets defined by all Banks sent to the IOT have been imaged and delivered to the output destination; the IOT has processed the Bank specifying CycleDown, and has cycled down. The previous task state, while cycled up and during cycle down, will have been TaskReadyForRestart. The TaskComplete state will also be entered following a PspMetaReset, a PspSheetBankAbort-AllBanksAbort, or a PspSheetBankAbort-ThisJobAbort, if it is the only Bank remaining to be processed.

TaskInProcess The IOT is processing Banks which it has received via PspNextBankRequest and is printing normally. When a job is completed by processing a Bank specifying EndOfJob, TaskInProcess transitions to TaskReadyForRestart when cycle down begins. If the IOT receives a PspRequestStateChange-CycleDown during TaskInprocess, this substate will be maintained until the cycle down is completed, and will then transition to TaskIncomplete.

TaskReadyForRestart This substate is entered when the IOT, in the CycledUpPrinting machine state, processes a Bank specifying EndOfJob, i.e., the last sheet of the last Bank has been printed and the IOT is delivering the previously imaged sheets to their destination. If this state is entered because the IOT has processed the Bank specifying EndOfJob, it will transition to TaskComplete when the cycle down is finished, including delivery of the the last sheet of the last copy from a finisher.

TaskIncomplete This substate will be entered from TaskInProcess if the IOT completes a cycle down without completely processing the Bank(s) it has been given. This may be due to an interrupted job or to extended processing in a finisher.

FaultNotDetected The machine has no detected faults which prevent procressing the current job. This substate may exist in any of the machine states.

FaultDetected A fault which prevents processing the current job has been detected, and some action by either the Tech Rep or the Operator, or both, is required. The current job is the job which was in process and is now incomplete, or a job that has been requested via PspNextBankRequest. The machine will usually be in the CyclingDown or CycledDownNotReady state when this substate is reported. However, the machine may be cycled up for the purpose of running diagnostics while this substate exists. The FaultDetected substate always induces the nonproductive substate, as well. The details of a fault state must be reported to the PSP via TechRepClearOnlyFaults, OperatorClearOnlyFaults, or TechRepRetryFaults of IotOperationalInfo. There may be certain detected faults which do not prevent the current job from being procressed. Such faults must be reported via IotOperationalInfo as FaultWarnings in the HintMessage field, rather than via IotStateInfo.

Productive The IOT is processing customer's jobs, or is available to process customer's jobs. Note that the IOT may also be performing diagnostics which do not interfere with the processing of customer's jobs. The diagnostic activity must be reported to the PSP via the HintMessage or InfoMessage of IotOperationalInfo.

NonProductive The IOT is not available for customer's jobs. The IOT has detected a fault, as described previously, and/or it is performing some nonproductive activity, such as processing diagnostic jobs, performing process-control, set-up, or calibration. The details of such nonproductive activity must be reported to the PSP via the HintMessage or InfoMessage of IotOperationalInfo.

The machine states are defined as follows:

CycledDownStandby The slow-scan mechanism is not running, the task substate is TaskComplete or TaskIncomplete, there are no detected faults, and the Productivity substate is Productive. TaskComplete implies that there is no longer any eventuality which would require re-imaging any sheet of the previous job. The IOT is ready to be cycled up for normal printing. In this machine state, TaskIncomplete implies that the PSP, for some reason, ordered cycle down, or the IOT, for internal reasons, initiated cycle down before the job in process was completed. Note that this machine state may change to CycledDownNotReady, if a new job is programmed that specifies a facility which is not ready.

CycledDownNotReady The slow-scan mechanism is not running, the task substate is TaskComplete or TaskIncomplete, the IOT is in the NonProductive substate, and the fault substate may be FaultDetected or FaultNotDetected. The machine state may be caused to transition from CycledDownStandby to CycledDownNotReady, or vice versa, for example, by the detection of a fault or clearing of a fault, respectively, while cycled down, or through invoking the NonProductive substate, or returning to the Productive substate, respectively, while the machine is cycled down.

CyclingDown This is a transient state and in some IOTs may be included in the cycled up states. Reporting of this state is, therefore, optional. The slow-scan mechanism is being stopped after having been in the CycledUpPrinting or CycledUpNotReady machine state. Cycle down may be scheduled via PspNextBankRequest, or may be initiated via PspRequestIotStateChange or by internal IOT action. If the CyclingDown machine state is entered from CycledUpPrinting, the task substate, TaskInProcess or TaskReadyForRestart, will be maintained until the cycle down is completed, at which time the substate will transition to TaskIncomplete or TaskComplete, respectively. Note that when extended finishing is involved, the transition to TaskComplete must be delayed until the last sheet is delivered to the final destination. If the cycle-down is due to a fault, the fault substate will be FaultDetected. If the substates of CyclingDown are TaskReadyForRestart and FaultNotDetected, the IOT may be transitioned to machine state CyclingUp by receipt of PspRequestIotStateChange-CycleUp or PspNextBankRequest describing another job.

CyclingUp This is a transient state and in some IOTs may be included in the cycled down states. Reporting of this state is, therefore, optional. The slow-scan mechanism is being started up after having been in the CycledDownStandby or CycledDownNotReady machine state. Cycle-up from

CycledDownStandby is normally initiated by PspRequestIotStateChange. The normal substates during CyclingUp are TaskInProcess, FaultNotDetected, and Productive. The IOT may enter CyclingUp from CycledDownNotReady only if the Productivity substate is Productive. This may be initiated by internal IOT action. The IOT may also enter CyclingUp from CyclingDown, as described above under CyclingDown. The IOT may be transitioned from CyclingUp to CyclingDown by receipt of PspRequestIotStateChange-CycleDown.

CycledUpPrinting The slow-scan mechanism is running, the IOT is processing Banks, and output is being delivered. The normal substates of this machine state are TaskInProcess or TaskReadyForRestart, FaultNotDetected, and Productive. The IOT normally transitions to CyclingDown after processing a Bank with the EndOfJob flag set. The IOT may also enter CyclingDown upon receiving PspRequestIotAction-cycledown, or by internal action upon detection of a fault.

CycledUpNotReady The slow-scan mechanism is running, the IOT is in the NonProductive substate, or in an extended period of dead cycling between jobs. The NonProductive substate indicates that some nonproductive activity is taking place which precludes printing a customer job, e.g., process control, calibration, set-up, etc. The Task substates may be TaskComplete or TaskIncomplete, depending on the conditions existing when the IOT entered this machine state. The normal fault substate is FaultNotDetected; however, FaultDetected may exist if it does not preclude the cycled-up activity.

IOTOPERATIONALINFO (88)

PARAMETERS InformationType (1) DataField (6, or variable 1 to 122)

ORIGINATOR IOT

FUNCTION To supply information in response to PspReadIotOperationalInfo, or voluntarily.

TRANSMISSION ORDER:

	ID CODE	Information Type	DataField		DataField		DataField
Byte Nos.	B0	B1	B2	— —	B7	— —	B123

DEFINITION:

IOTOPERATIONALINFO ::= IDCODE InformationType DataField

IDCODE ::= B0 -- byte 0

B0 ::= 88 hex

InformationType DataField ::= B1 B2B3B4B5B6B7 .. B123

B1 B2B3B4B5B6B7 .. B123 ::= UNDEFINED
| TECHREPCLEARONLYFAULTS NumberOfFaults Spare FaultCodes
| OPERATORCLEARFAULTS NumberOfFaults Spare FaultCodes
| TECHREPRETRYFAULTS NumberOfFaults Spare FaultCodes
| HINTMESSAGE NumberOfMessages Spare MessageCodes
| INFOMESSAGE NumberOfMessages Spare MessageCodes
| FEEDER0 FeederInfo
| FEEDER1 FeederInfo
| FEEDER2 FeederInfo
| FEEDER3 FeederInfo
| FEEDER4 FeederInfo
| FEEDER5 FeederInfo
| FEEDER6 FeederInfo
| FEEDER7 FeederInfo
| DESTINATION0 DestinationInfo
| DESTINATION1 DestinationInfo
| DESTINATION2 DestinationInfo
| DESTINATION3 DestinationInfo
| DESTINATION4 DestinationInfo
| DESTINATION5 DestinationInfo
| DESTINATION6 DestinationInfo
| DESTINATION7 DestinationInfo
| CRASHRECOVERYSTATUS LastMessage CrashInfo
| SPARE SpareInfo

UNDEFINED ::= 00 hex

TECHREPCLEARONLYFAULTS ::= 01 hex -- variable length format

NumberOfFaults ::= B2

B2 ::= [00 .. 3C] hex

Spare	::= B3
B3	::= [00 .. FF] hex -- not sent if number of faults =0
FaultCodes	::= B4B5 .. B(n-1)Bn -- IOT specific 2-byte numbers; n = (3 + 2 (number of faults)) for (number of faults) > 1, maximum = 60.
OPERATORCLEARFAULTS	::= 02 hex -- variable length format
NumberOfFaults	::= B2
B2	::= [00 .. 3C] hex
Spare	::= B3
B3	::= [00 .. FF] hex -- see comments at TechRepClearOnlyFaults
FaultCodes	::= B4B5 .. B(n-1)Bn
TECHREPRETRYFAULTS	::= 03 hex -- variable length format
NumberOfFaults	::= B2
B2	::= [00 .. 3C] hex
Spare	::= B3
B3	::= [00 .. FF] hex -- see comments at TechRepClearOnlyFaults
FaultCodes	::= B4B5 .. B(n-1)Bn
HINTMESSAGE	::= 04 hex -- variable length format
NumberOfMessages	::= B2
B2	::= [00 .. 3C] hex
Spare	::= B3
B3	::= [00 .. FF] hex -- see comments at TechRepClearOnlyFaults
MessageCodes	::= B4B5 .. B(n-1)Bn
INFOMESSAGE	::= 05 hex -- variable length format
NumberOfMessages	::= B2
B2	::= [00 .. 3C] hex
Spare	::= B3
B3	::= [00 .. FF] hex -- see comments at TechRepClearOnlyFaults
MessageCodes	::= B4B5 .. B(n-1)Bn
FEEDER0	::= 06 hex -- seven byte format
FEEDER1	::= 07 hex
FEEDER2	::= 08 hex
FEEDER3	::= 09 hex
FEEDER4	::= 0A hex
FEEDER5	::= 0B hex

FEEDER6	::=	0C hex
FEEDER7	::=	0D hex
FeederInfo	::=	TrayStatus TrayPaperGauge TrayPaperWidth TrayPaperLength
TrayStatus	::=	B2
B2	::=	NotAvailable | Ready | Selected | Lowering | Lowered | Jammed | Raising | Broken | Empty | Spare
NotAvailable	::=	00 hex
Ready	::=	01 hex
Selected	::=	02 hex
Lowering	::=	03 hex
Lowered	::=	04 hex
Jammed	::=	05 hex
Raising	::=	06 hex
Broken	::=	07 hex -- Not operator clearable.
Empty	::=	08 hex
Spare	::=	[09..FF] hex
TrayPaperGauge	::=	B3
B3	::=	TrayGaugeInfo
TrayGaugeInfo	::=	[00..FF] hex -- IOT specific
TrayPaperWidth	::=	B4B5
B4B5	::=	NotAvailable | WidthDimension
NotAvailable	::=	0000
WidthDimension	::=	[0001 .. FFFF] hex -- in mm.
TrayPaperLength	::=	B6B7
B6B7	::=	NotAvailable |LengthDimension
NotAvailable	::=	0000 hex
LengthDimension	::=	[0001 .. FFFF] hex -- in mm.
DESTINATION0	::=	0E hex -- seven byte format
DESTINATION1	::=	0F hex
DESTINATION2	::=	10 hex
DESTINATION3	::=	11 hex
DESTINATION4	::=	12 hex
DESTINATION5	::=	13 hex
DESTINATION6	::=	14 hex
DESTINATION7	::=	15 hex

DestinationInfo	::=	DestinationStatus DestinationPaperGauge DestinationPaperWidth DestinationPaperLength
DestinationStatus	::=	B2
B2	::=	NotAvailable | Ready | Selected | Lowering | Lowered | Jammed | Raising | Broken | Full | ReadyEmpty | Spare
NotAvailable	::=	00 hex
Ready	::=	01 hex
Selected	::=	02 hex
Lowering	::=	03 hex
Lowered	::=	04 hex
Jammed	::=	05 hex
Raising	::=	06 hex
Broken	::=	07 hex
Full	::=	08 hex
ReadyEmpty	::=	09 hex
Spare	::=	[0A .. FF] hex
DestinationPaperGauge	::=	B3
B3	::=	DestinationGaugeInfo
DestinationGaugeInfo	::=	[00 .. FF] hex -- IOT specific
DestinationPaperWidth	::=	B4B5
B4B5	::=	Not Available | WidthDimension
NotAvailable	::=	0000
WidthDimension	::=	[0001 .. FFFF] hex -- in mm.
DestinationPaperLength	::=	B6B7
B6B7	::=	Not Available | LengthDimension
NotAvailable	::=	0000
WidthDimension	::=	[0001 .. FFFF] hex -- in mm.
CrashRecoveryStatus	::=	16 hex
LastMessage	::=	B2
B2	::=	No | Yes | Unused
No	::=	00 hex
Yes	::=	01 hex
Unused	::=	[02 .. FF] hex
CrashInfo	::=	JobNumber JobState
JobNumber	::=	B3
B3	::=	Undefined | JobIdentifier
Undefined	::=	00 hex

JobIdentifier	::= [01 .. FF] hex
JobState	::= B4 [B5 .. B9] -- 1 or 6 bytes long
B4 [B5 .. B9]	::= Complete | Incomplete Hint
Complete	::= 00 hex
Incomplete	::= 01 hex
Hint	::= PlateNumber SheetNumber CopyNumber
PlateNumber	::= B5
B5	::= PageMode PlateColor PlateResolution
PageMode	::= d1 d0
d1d0	::= Duplex | Simplex | SimultaneousDuplex | Spare1
Duplex	::= 00 binary
Simplex	::= 01 binary
SimultaneousDuplex	::= 10 binary
Spare1	::= 11 binary
PlateColor	::= d5 d4 d3d2
PlateDeadCycle	::= 000000 binary
d2	::= NotColor0 | Color0
NotColor0	::= 0 binary
Color0	::= 1 binary
d3	::= NotColor1 | Color1
NotColor1	::= 0 binary
Color1	::= 1 binary
d4	::= NotColor2 | Color2
NotColor2	::= 0 binary
Color2	::= 1 binary
d5	::= NotColor3 | Color3
NotColor3	::= 0 binary
Color3	::= 1 binary
ResolutionMode	::= d7 d6
d6	::= SimplexRes1 | SimplexRes2
SimplexRes1	::= 0 binary -- IOT specific
SimplexRes2	::= 1 binary -- IOT specific
d7	::= DuplexRes1 | DuplexRes2
DuplexRes1	::= 0 binary -- IOT specific
DuplexRes2	::= 1 binary -- IOT specific
SheetNumber	::= B6 B7
B6 B7	::= SheetDeadCycle | SheetIdentifier

	SheetDeadCycle	::=	0000 hex
	SheetIdentifier	::=	[0001 .. FFFF] hex
	CopyNumber	::=	B8 B9
	B8 B9	::=	SampleCopy | CopyIdentifier
	SampleCopy	::=	0000 hex
	CopyIdentifier	::=	[0001 .. FFFF] hex
SPARE		::=	[17 .. FF] hex

IN RESPONSE TO PSPREADIOTOPERATIONALINFO Volunteered

APPLICATION NOTES

The five categories of InformationType give rise to three different formats for these messages.

The Fault and Message information types require from 2 to 123 bytes (excluding ID Code), depending on the number of fault codes or message codes to be transmitted:

Byte Nos.	B0	B1	B2	B3	B4 B5		B122 B123
	ID CODE	(Type of Fault or Message)	(No. of Faults or Messages)	Spare	(Highest priority Fault or Message code)	— —	(Lowest priority Fault or Message code)

All fault (or message) codes in a particular category are included, in order of priority, in the data field of a single message. The highest priority is transmitted first. The data field will accommodate up to sixty fault (or message) codes. It is the responsibility of the IOT to prioritize the faults (or messages). If there are fault (or message) codes in different categories, they are sent in sequential messages in category code order. If there are no fault (or message) codes, byte B2 is 00 hex and the format is truncated following B2.

(Also refer to section 6.4.8, "Fault Protocol.")

FEEDER and DESTINATION information types each require 7 bytes (excluding ID Code):

	ID Code	FeederN	TrayStatus	TrayPaper Gauge	TrayPaperWidth		TrayPaperLength	
Byte Nos.	B0	B1	B2	B3	B4	B5	B6	B7

	ID Code	DestinationN	Destination Status	Destination PaperGauge	DestinationPaperWidth		DestinationPaperLength	
Byte Nos.	B0	B1	B2	B3	B4	B5	B6	B7

The CRASHRECOVERYSTATUS type requires 4 or 9 bytes (excluding ID Code):

	ID Code	CrashRecoveryStatus	LastMessage	JobNumber	JobState	PlateNumber	SheetNumber		CopyNumber	
Byte Nos.	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9

After receiving a PspReadIotOperationalInfo message for CrashRecoveryStatus, the IOT will respond with one or more IotOperationalInfo messages of type CrashRecoveryStatus. Each message in this series will describe the crash recovery state of one job. The last message in this series will have "LastMessage" set to "Yes," which is value 01 hex. For each message, if the job is complete, the JobState will be "Complete" (00 hex) and that will be the last byte of the message. If the job's state is incomplete, the JobState byte will be Incomplete (01 hex). Following this Incomplete state byte will be the Advance Video Hint for that job. This hint is the PlateNumber/SheetNumber/CopyNumber that the IOT would hint for that job if the IOT were requested to cycle up at that time. While the SheetNumber and CopyNumber offer possible selection of SheetDeadCycle and SampleCopy respectively, these are assigned for completeness and are never expected during the Crash Recovery exchange. The IOT should never volunteer this CrashRecoverStatus messge without first receiving a PSP request for this info. The PSP is allowed to request this info only between the time that the IOT reports a state info message after configuration exchange and before the first cycle up request. The PSP is not required to ask for the Crash Recovery exchange. The length of the job history that the IOT must keep is IOT specific. Some IOTs might be able to send a subset of the total job history in order to optimize Crash Recovery. This optimization is program specific.

The IOT may indicate either that it knows of no jobs, or was not able to recover any jobs, by sending the following response:

> [IotOperationalInfo (88h), CrashRecoveryStatus (16h), LastMessage: Yes (01h), CrashInfo: [JobNumber: 00h, JobState: Complete (00h)]]

When the IOT sends this response, the PSP will assume that any jobs that it thinks are incomplete need to be remade.

IOTDIAGNOSTICRESPONSE (89)

PARAMETERS Status (1) Utility (2) ParameterID (2) ParameterValue (2)

ORIGINATOR IOT

FUNCTION Returns the response to the PSP, for the requested diagnostic program.

TRANSMISSION ORDER:

	ID Code	Status	Utility		ParameterID		ParameterValue	
Byte Nos.	B0	B1	B2	B3	B4	B5	B6	B7

DEFINITION:

IOTDIAGNOSTICRESPONSE	::= IDCODE Status Utility ParameterID ParameterValue
IDCODE	::= B0 --byte 0
B0	::= 89 hex
Status	::= B1
B1	::= Enter | Start | Stop |Exit | Read | Write | Info | Reject | Spare
Enter	::= 00 hex -- Response to Enter request.
Start	::= 01 hex -- Response to Start request.
Stop	::= 02 hex -- Response to Stop request.
Exit	::= 03 hex -- Response to Exit request.
Read	::= 04 hex -- Response to Read request.
Write	::= 05 hex -- Response to Write request.
Info	::= 06 hex -- Volunteered
Reject	::= 07 hex -- Specific request rejected.
Spare	::= [08 .. FF] hex
Utility	::= B2B3
B2B3	::= [00 .. FF] hex
ParameterID	::= B4B5
B4B5	::= [0000 .. FFFF] hex -- Set of parmeters unique to utility Status.
ParameterValue	::= B6B7
B6B7	::= [0000 .. FFFF] hex -- Set of parameter values.

IN RESPONSE TO PSPREQUESTIOTDIAGNOSTIC Command Utility ParameterID ParameterValue

APPLICATION NOTES This message is defined so that when responding to PspRequestIotDiagnostic, it will, in general, echo the Utility and Command designations contained in the PspRequestIotDiagnostic message. For *Write*, the ParameterID and ParameterValue are also echoed. For *Read*, the ParameterID is echoed, and the ParameterValue is supplied. For *Reject*, all fields are echoed except the Command designation. When not applicable, the ParameterID and/or ParameterValue fields should be 0000 Hex. Also refer to paragraph 6.6.

IOTREJECTPSPCOMMAND (8B)

PARAMETERS IotReason (1) PspCommandRejected (1) PspCommandParameterNumber (1) SheetNumber (2) CopyNumber (2) JobNumber (1) Spare (1) PspCommandParameterValue (variable 1 to 4)

ORIGINATOR IOT

FUNCTION To inform the PSP that the command sent to the IOT is not valid at the time of receipt.

TRANSMISSION ORDER:

	ID Code	IotReason	PspCommand Rejected	PspCommand ParamNumber	SheetNumber	
Byte Nos.	B0	B1	B2	B3	B4	B5

	CopyNumber		JobNumber	Spare	PsPCommandParameterValue		
Byte Nos.	B6	B7	B8	B9	B10	— —	B13

DEFINITION:

IOTREJECTPSPCOMMAND	::=	IDCODE IotReason PspCommandRejected PspCommandParameterNumber SheetNumber CopyNumber JobNumber Spare PspCommandParameterValue
IDCODE	::=	B0 --byte 0
B0	::=	8B hex
IotReason	::=	B1
B1	::=	UNDEFINEDCOMMAND | ILLEGALSEQUENCE | INVALIDCOMMANDPARAMETER | COMMANDBUFFERFULL | UNSUPPORTEDCOMMAND | INVALIDFORMAT | Spare
UNDEFINEDCOMMAND	::=	01 hex -- command ID Code not recognized as valid.
ILLEGALSEQUENCE	::=	02 hex -- receipt of valid command at an invalid IOT state for that command.
INVALIDCOMMANDPARAMETER	::=	03 hex -- PSP Command parameter number out of valid range.
COMMANDBUFFERFULL	::=	04 hex -- IOT too busy to handle any further request at this time.
UNSUPPORTEDCOMMAND	::=	05 hex -- command ID Code not implemented in this revision software.
INVALIDFORMAT	::=	06 hex -- too few or too many parameters for the valid ID code received.
Spare	::=	[07 .. FF] hex

PspCommandRejected	::=	B2
B2	::=	PSPCONFIGURATION | PSPNEXTBANKREQUEST | PSPPRINT | PSPREADIOTMEMORY | PSPREADIOTSTATE | PSPREADIOTSTATUS | PSPREQUESTIOTDIAGNOSTIC | PSPSHEETBANKABORT | PSPWRITEIOTMEMORY | PSPREQUESTIOTACTION | PSPREQUESTIOTSTATECHANGE
PSPCONFIGURATION	::=	01
PSPNEXTBANKREQUEST	::=	03
PSPPRINT	::=	04
PSPREADIOTMEMORY	::=	06
PSPREADIOTSTATE	::=	07
PSPREADIOTSTATUS	::=	08
PSPREQUESTIOTDIAGNOSTIC	::=	09
PSPSHEETBANKABORT	::=	0C
PSPWRITEIOTMEMORY	::=	0D
PSPREQUESTIOTACTION	::=	0E
PSPREQUESTIOTSTATECHANGE	::=	0F
PspCommandParameterNumber	::=	B3
B3	::=	[00 .. FF] hex -- number of the first command parameter sent by the PSP which was found to be in error.
SheetNumber	::=	B4 B5
B4 B5	::=	[NoApplicable | SheetIdentifier
NotApplicable	::=	0000
SheetIdentifier	::=	[0001 .. FFFF] hex
CopyNumber	::=	B6 B7
B6 B7	::=	NotApplicable | CopyIdentifier
NotApplicable	::=	0000
CopyIdentifier	::=	[0001 .. FFFF] hex
JobNumber	::=	B8
B8	::=	NotApplicable | JobIdentifier
NotApplicable	::=	0000
JobIdentifier	::=	[0001 .. FFFF] hex
Spare	::=	B9
B9	::=	[00 .. FF] hex
PspCommandParameterValue	::=	[B10 .. B13]
[B10 .. B13]	::=	[00 .. FFFFFFFF] hex --Command parameter value sent by the PSP.

IN RESPONSE TO Volunteered, PSP response to be product specific.

APPLICATION NOTES *IllegalSequence* includes the following cases:

The IOT receives PspRequestIotAction/CycleUp, but has not properly received or stored the necessary job Bank. The IOT will reject the command, specify *illegalSequence*, and designate ChangeRequested as the PspCommandParameter (parameter no. 1), and the CycleUp code as the PspCommandParameterValue.

The PSP attempts to program too many Banks. The IOT will reject PspNextBankRequest, specify *illegalSequence*, specify the appropriate SheetNumber and JobNumber parameters, specify CopyNumber as not applicable, and designate parameter no. 19 (JobNumber) as the PspCommandParameter, and the numerical job number as the PspCommandParameterValue. Note that parameter numbering is not the same as byte numbering (refer to section 3.2.1).

When *Invalid Format* is specified, the entry for PspCommandParameterNumber should be as follows:

If there are too few parameters, enter the number of the first missing parameter, i.e., the actual number of parameters received plus one.

If there are too many parameters, enter the number of the first excess parameter, i.e., the expected number plus one.

(Refer to section 6.3.1 for convention on parameter numbering.)

SheetNumber, CopyNumber, and JobNumber should be specified as Not Applicable if they were not included in the command being rejected.

The PspCommandParameterValue field is variable, according to the length of the specific parameter value to be transmitted.

IOTSHEETDELIVERED (8C)

PARAMETERS Integrity (1) SheetNumber (2) CopyNumber (2) Destination (1) SorterBin (1) JobNumber (1)

ORIGINATOR IOT

FUNCTION To inform PSP when each sheet has been successfully delivered to an output area, or whenever a scratch sheet has been delivered to an output area.

TRANSMISSION ORDER:

	ID CODE	Integrity	SheetNumber1		CopyNumber		Destination	SorterBin	Job Number
Byte Nos.	B0	B1	B2	B3	B4	B5	B6	B7	B8

DEFINITION:

```
IOTSHEETDELIVERED          ::= IDCODE Integrity SheetNumber CopyNumber
                               Destination  SorterBin  JobNumber
  IDCODE                   ::= B0   -- byte 0
    B0                     ::= 8C hex
  Integrity                ::= B1
    B1                     ::= GOODSHEET | SCRATCHSHEET | SPARE
      GOODSHEET            ::= 01 hex
      SCRATCHSHEET         ::= 02 hex
      SPARE                ::= [03 .. FF] hex
  SheetNumber              ::= B2B3 --   Sheet delivered to destination.
    B2B3                   ::= Undefined | SheetIdentifier
      Undefined            ::= 0000 hex
      SheetIdentifier      ::= [0001 .. FFFF] hex
  CopyNumber               ::= B4B5
    B4B5                   ::= Sample | SheetIdentifier
      Sample               ::= 0000 hex
      SheetIdentifier      ::= [0001 .. FFFF] hex
  Destination              ::= B6
    B6                     ::= DESTINATION0 | DESTINATION1
                               | DESTINATION2
                               | DESTINATION3 | DESTINATION4
                               | DESTINATION5
                               | DESTINATION6 | DESTINATION7 | SPARE
                               | DUPLEXTRAY
      DESTINATION0         ::= Intermediate | Final
        Intermediate       ::= 80 hex
        Final              ::= 00 hex
      DESTINATION1         ::= Intermediate | Final
        Intermediate       ::= 81 hex
        Final              ::= 01 hex
      DESTINATION2         ::= Intermediate | Final
        Intermediate       ::= 82 hex
        Final              ::= 02 hex
      DESTINATION3         ::= Intermediate | Final
```


Intermediate	::=	83 hex
Final	::=	03 hex
DESTINATION4	::=	Intermediate | Final
Intermediate	::=	84 hex
Final	::=	04 hex
DESTINATION5	::=	Intermediate | Final
Intermediate	::=	85 hex
Final	::=	05 hex
DESTINATION6	::=	Intermediate | Final
Intermediate	::=	86 hex
Final	::=	06 hex
DESTINATION7	::=	Intermediate | Final
Intermediate	::=	87 hex
Final	::=	07 hex
SPARE	::=	Intermediate | Final
Intermediate	::=	[88 .. 8F] hex
Final	::=	[08 .. 7F] hex
DUPLEXTRAY	::=	FF hex
SorterBin	::=	B7
B7	::=	Undefined | SorterBinNumber
Undefined	::=	00 hex
SorterBinNumber	::=	[01 .. FF] hex
JobNumber	::=	B8
B8	::=	Undefined | JobIdentifier
Undefined	::=	00 hex
JobIdentifier	::=	[01 .. FF] hex

IN RESPONSE TO Volunteered

APPLICATION NOTES

If PspConfiguration designates ReturnOutputSheetNumber-OfEachSheet, and an intermediate finishing destination is involved, the 8x series of hex destination codes will be returned as each sheet is delivered to the intermediate destination. When the set is ejected to the final destination, the 0x series of destination codes will be returned in conjunction with the last sheet number only. It is to be understood that whenever sheet number is required, the copy number must also be returned.

IOTREQUESTMEMORY (8D)

PARAMETERS Spare (1) StartAddress (4) EndAddress (4)

ORIGINATOR IOT

FUNCTION To request that the PSP down load the designated segment of the IOT's software which is stored in the PSP's non-volatile memory, via PspWriteIotMemory commands.

TRANSMISSION ORDER:

	ID CODE	Spare	StartAddress				EndAddress			
Byte Nos.	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9

DEFINITION:

IOTREQUESTNONVOLATILEMEMORY	::= IDCODE Spare StartAddress EndAddress
IDCODE	::= B0 --byte 0
B0	::= 8D hex
Spare	::= B1
B1	::= [00 .. FF] hex
StartAddress	::= B2B3B4B5
B2B3B4B5	::= [00000000 .. FFFFFFFF] hex
EndAddress	::= B6B7B8B9
B6B7B8B9	::= [00000000 .. FFFFFFFF] hex

IN RESPONSE TO Volunteered

APPLICATION NOTES The IOT may utilize non-volatile memory, which is physically located in the PSP. This message is used to initiate downloading.

IOTSWITCHINFO (8E)

PARAMETERS Switch (1) Action (1)

ORIGINATOR IOT

FUNCTION To inform the PSP that the state of an external IOT switch has changed due to operator action.

TRANSMISSION ORDER:

	ID CODE	Switch	Action
Byte Nos.	B0	B1	B2

DEFINITION:

IOTSWITCHINFO	::= IDCODE Switch Action
IDCODE	::= B0 --byte 0
B0	::= 8E hex
Switch	::= B1
B1	::= START | STOP | SAMPLE | SPARE | TRAYS | BINS
START	::= 01 hex
STOP	::= 02 hex
SAMPLE	::= 03 hex
SPARE	::= [04 .. 1F] hex
TRAYS	::= [20 .. 2F] hex
BINS	::= [30 .. FF] hex
Action	::= B2
B2	::= ON | MOMENTARYON | UNDEFINED | OFF | MOMENTARYOFF | UNDEFINED
ON	::= 00 hex
MOMENTARYON	::= 01 hex
UNDEFINED	::= [02 .. 7F] hex
OFF	::= 80 hex
MOMENTARYOFF	::= 81 hex
UNDEFINED	::= [82 .. FF] hex

IN RESPONSE TO Volunteered

6.4 Operational protocol

6.4.1 IOT states

The generic states of the IOT are illustrated in figure 6-3. The machine states, as well as substates, are defined in the Application Notes for IotStateInfo. An IOT may have more internal states, but they must fall within the defined states.

The two normal operating states are CycledUpPrinting and CycledDownStandby. Two additional steady states, CycledUpNotReady and CycledDownNotReady, are prescribed to account for exception conditions. Because some IOTs will require a significant time to transition between the CycledUp and CycledDown states, two transient states, CyclingUp and CyclingDown, are also prescribed. However, separate reporting of these states by the IOT is optional. CyclingUp may be included as the end of CycledDownStandby (or CycledDownNotReady), and CyclingDown may be included as the end of CycledUpPrinting (or CycledUpNotReady). Note that there are no distinct diagnostic modes. Diagnostics may be run in any of the machine states. The Productivity substate may be *Productive* or *NonProductive*, depending on whether or not the diagnostic utility renders the IOT unavailable for customer jobs.

The normal state transitions are from CycledDownStandby to CycledUpPrinting, and vice versa. If a fault occurs in CycledDownStandby, or if an activity is invoked which requires the NonProductive substate, the machine state transitions to CycledDownNotReady. (The NonProductive substate is required by some diagnostics and by non-job activity, such as process control, calibration, set-up, etc.) Similarly in CycledUpPrinting, if the NonProductive substate is invoked, the machine state transitions to CycledUpNotReady. If due to a fault, the machine would normally continue on to CycledDownNotReady. If due to other non-job activity, the machine state could transition back to CycledUpPrinting when the NonProductive substate is terminated. The IOT may be transitioned between CycledDownNotReady and CycledUpNotReady for service purposes, provided that a fault which precludes cycling up does not exist. The task substates apply only to print jobs, customer or diagnostic, not to the non-job activities.

6.4.2 Message sequencing and timing requirements

The printing of each image is accomplished by a triplet of Command and Status messages and a cycle of **page sync**. The messages are IotVideoHint (Hint), by which the IOT indicates to the PSP which image it expects next; PspPrint (Print), by which the PSP orders the IOT to schedule printing of the indicated image (usually the one expected by the IOT); and IotVideoRequest (Request), by which the IOT requests the video signal for the image designated in the Print command. For a given image, the messages occur in the sequence above, but within a page time they will bear different image indexes and must occur in the sequence, and according to the timing relationships shown in figure 6-4. *Page-time* is defined as the interval between successive positive transitions of **page sync**. (An IOT may implement more than one page-pitch, with corresponding page-times. The message timing requirements may be met by referencing each case to the respective page-time, or they may be met for all cases by using the timing derived from the minimum page-time.) Either IotVideoRequest or IotVideoHint may be the first message within a given page-time, but both must occur in the first 20 percent of the page-time. IotVideoRequest must occur during the page-time preceding the page-time of the image being requested, so as to provide sufficient time for the PSP to make the appropriate image available at the interface. (Note that the video for the requested image is delivered beginning with the next positive transition of **page sync**.) The Request is accompanied by a Hint for a subsequent image, and then a Print for that subsequent image. Print must occur within the first 85 percent of the page-time, always following the Hint. The sheet number index of the Hint and Print messages must be one or more greater than the index of the Request occurring during the same page-time. The index difference, x, is the *scheduling-offset,* an integer equal to or greater than one, which represents the number of **page sync** positive transitions which must occur between the Hint (or Print) and Request for the same image. The scheduling-offset is the larger of the unique scheduling-offset requirements of the PSP and the IOT. The PSP will have a scheduling-offset requirement to allow for worst case retrieval of an image. The PSP must determine this offset from knowledge of the IOT paper path lengths (simplex and duplex), and from the PSP's resource management burden. The IOT will have a scheduling-offset requirement to allow for worst case control processing, imaging, and paper feeding to be completed for a given image *after* the Print command is received for that image by the IOT. The PSP and IOT exchange these requirements via PspConfiguration and IotConfiguration, respectively, and then each independently selects the larger for the operational scheduling-offset. The paper path information needed by the PSP is also conveyed by IotConfiguration.

Note that although page-time is defined with reference to page sync and the message timing is referenced to page-time, a page sync true interval may not always be present at the physical interface for every message sequence (refer to section 6.4.4, Page Sync Regimen).

The timing of PspNextBankRequest (NBR) and PspSheetBankAbort (SBA) is also specified in figure 6-3. These are non-repetitive messages required for initiating, controlling, and terminating the imaging process. NBR must occur no later than 0.3 of a page-time preceding onset of the page-time during which the Print command will occur for the image on which the next bank is to become effective. It may occur any time earlier.

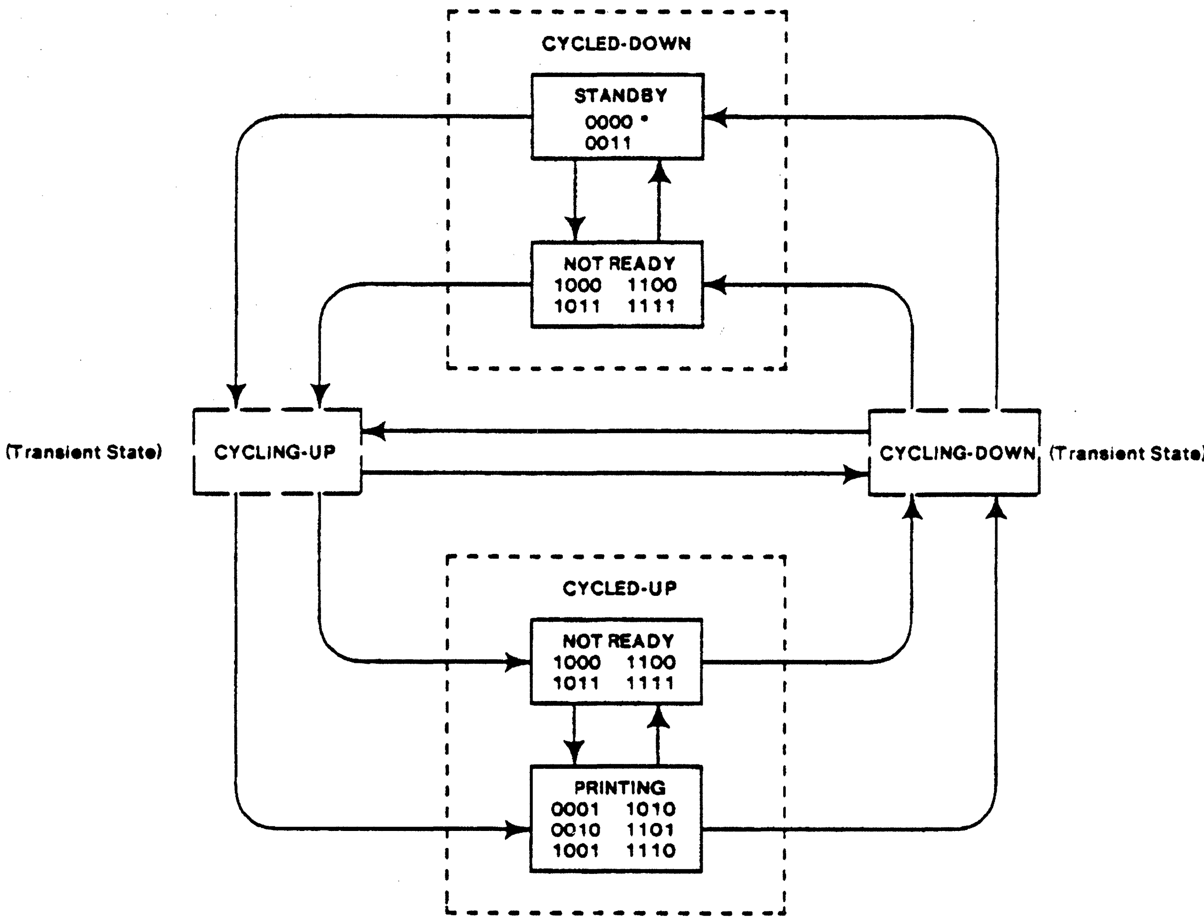
6.4.3 Numbering and sequencing conventions

Four parameters are used in the numbering of sheets and images: PlateNumber, SheetNumber, CopyNumber, and JobNumber. JobNumber is used to distinguish between duplicate SheetNumbers which are caused by job interrupts (refer to section 6.4.6). The following discussion applies to non-interrupt conditions. It is understood that during interrupt conditions, JobNumber must be used in conjunction with SheetNumber to guarantee uniqueness. SheetNumber designates a particular sheet of a copy-set. CopyNumber designates the replication of sheets with the same image(s), or the replication of a set of sheets. PlateNumber is a byte code in which 2 bits denote which side of the sheet (simplex/duplex) is to be imaged, 4 bits designate in which of four colors the image is to be rendered, and 2 bits designate one of two resolutions for each side of the sheet. Thus, the SheetNumber plus CopyNumber[1] plus the simplex/duplex bits uniquely designate an image, the chroma bits designate a subset of that image, and the resolution bits designate the resolution of that image. SheetNumber plus CopyNumber[1] plus PlateNumber, therefore, designates a unique plate. Also, SheetNumber plus CopyNumber designates a particular sheet of paper in the IOT. These relationships are shown graphically in figure 6-5 (also refer to the Glossary). SheetNumber zero is reserved to designate a dead cycle, i.e., a page-time of the IOT during which imaging (and page sync) is normally suspended. (Refer to section 6.4.4 for exceptions.) CopyNumber zero is reserved to designate a sample copy. When there is no ambiguity implied, the convenient terms "image number" and "sheet number" are used to mean the designation of a unique image or a unique sheet, respectively. (Note that the term *PlateMode* rather than PlateNumber is used in PspNextBankRequest, because in that instance, the information applies to more than one particular plate.)

Jobs are numbered consecutively in ascending order, modulo 256, excluding zero. Zero is reserved to designate "all jobs" in PspSheetBankAbort. JobNumbers may exist concurrently within the IOT to designate *previous* jobs, *current* jobs, *next* jobs, and *interrupted* jobs. A previous job is one which has completed imaging, but not yet completed delivery to the final destination. There can be more than one previous job. (The implication is that they are relatively small jobs co-existing in the output paper path.) Current jobs are jobs being imaged. There can be more than one current job, for example, when a new job is started while a duplex job is still processing the duplex side of its last sheet(s). A next job is a job, some or all of whose parameter banks have been programmed into the IOT during a current job and is, therefore, waiting to be imaged. There can be more than one next job. An interrupted job is the saved banks of a job which has suffered an ASAP interrupt during imaging (refer to section 6.4.6). There can be more than one interrupted job, i.e., nested interrupts. The IotConfiguration message specifies the maximum number of interrupted jobs and the maximum number of jobs of all four categories combined which the IOT can manage. The actual capability to be implemented by an IOT is a design issue affecting printing productivity. For example, a succession of one page jobs implies a number of *previous* jobs within the output paper path of an IOT, and a number of next jobs programmed into the IOT, if dead cycling is to be avoided.

[1] Copy Number is required to establish uniqueness only if Change Modification Entries are used.

Figure 6-3. **Generic IOT machine states**

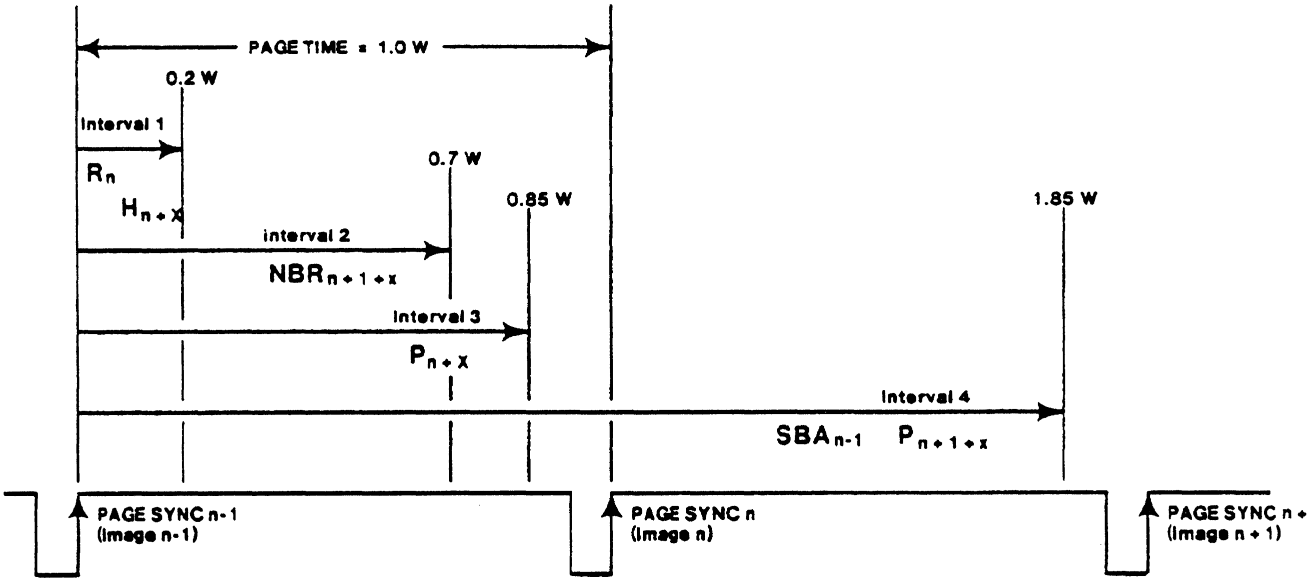


* Substate Codes [d3 .. d0] of byte 2 of IotStateInfo

Valid Substate Codes

Productivity d3	Fault d2	Task d1	Task d0			
0	0	0	0	Productive,	FaultNotDetected,	TaskComplete
0	0	0	1	Productive,	FaultNotDetected,	TaskInProcess
0	0	1	0	Productive,	FaultNotDetected,	TaskReadyForRestart
0	0	1	1	Productive,	FaultNotDetected,	TaskIncomplete
1	0	0	0	NonProductive,	FaultNotDetected,	TaskComplete
1	0	0	1	NonProductive,	FaultNotDetected,	TaskInProcess
1	0	1	0	NonProductive,	FaultNotDetected,	TaskReadyForRestart
1	0	1	1	NonProductive,	FaultNotDetected,	TaskIncomplete
1	1	0	0	NonProductive,	FaultDetected,	TaskComplete
1	1	0	1	NonProductive,	FaultDetected,	TaskInProcess
1	1	1	0	NonProductive,	FaultDetected,	TaskReadyForRestart
1	1	1	1	NonProductive,	FaultDetected,	TaskIncomplete

Figure 6-4. **Timing windows for PSP commands and IOT status messages**



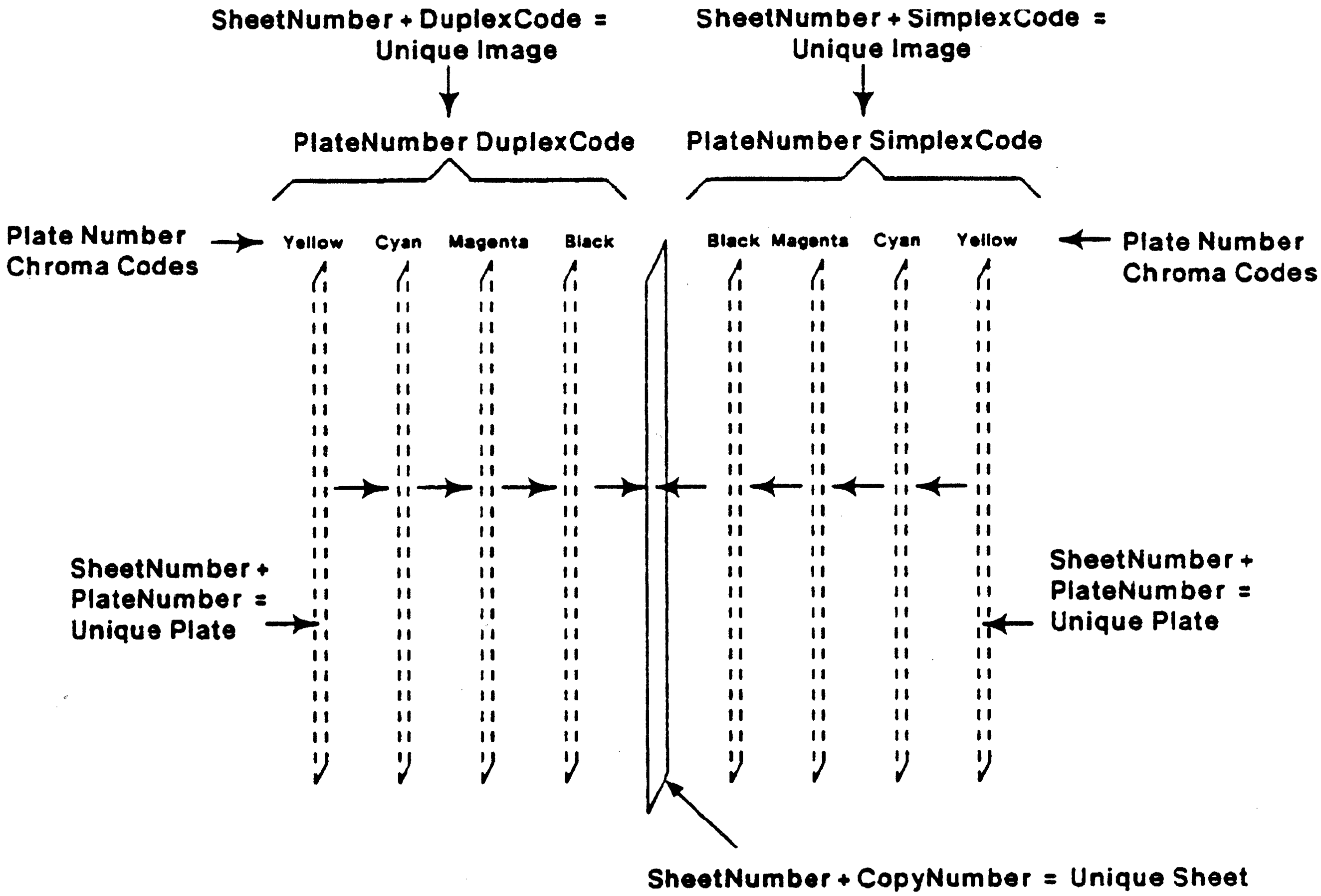
CYCLIC MESSAGES
- R_j = **IotVideoRequest j**
- H_j = **IotVideoHint j**
- P_j = **PspPrint j**

OCCASIONAL MESSAGES
- NBR_j = **PspNextBankRequest j**
- SBA_j = **PspSheetBankAbort j**

x = Scheduling Offset, ≥ 1

R_n and H_{n+x} must occur in interval 1, but may occur in either order within this interval.
P_{n+x} must occur in interval 3, and must follow R_n and H_{n+x}
NBR_{n+1+x} may occur in interval 2, or any time earlier.
SBA_{n-1} may occur in interval 4, or any time earlier, but must occur prior to P_{n+1+x}

Figure 6-5. **Numbering convention**



- **SheetNumber plus CopyNumber identifies unique sheet of paper**
- **SheetNumber plus -plexCode of PlateNumber identifies unique image.***
- **SheetNumber plus PlateNumber identifies unique plate.***

(JobNumber must be included with SheetNumber to insure uniqueness under job InterruptASAP conditions.)

*** CopyNumber is also required if Change Modification Entries are used.**

Sheets are numbered consecutively in ascending order, modulo 65,536, excluding zero. Zero is reserved to designate "dead cycle" in the imaging messages. SheetNumbers run consecutively from job to job. The consecutive rule is abrogated when SheetNumbers are reused in conjunction with CopyNumber in multicopy jobs, and in conjunction with JobNumber for interrupt jobs. (refer to section 6.4.6). The SheetNumbers of sheets within a collated set, or an uncollated stack, are always in ascending order, as delivered to the final destination. If SheetNumber j is the last sheet delivered in the previous job, then SheetNumber j+1 is the first sheet delivered in the current job, regardless of whether it is to be the first or last logical page of the output document. Image numbers (SheetNumber plus *-plex* bits) are also reused for multiplate images. SheetNumber may be reset between jobs which are separated by a normal cycle-down.

Jobs are characterized as *1-to-N* or *N-to-1*, according to the input/output sequencing used. These terms refer to the order in which the logical pages of a document are handled, assuming that they are numbered according to the normal English reading sequence. Jobs must be processed in 1-to-N order to output destination devices which stack paper face down, so as to maintain the proper page sequence. N-to-1 order must be used when the output is stacked face up. Job files may be delivered to the PSP in either 1-to-N or N-to-1 order. A PSP may reverse the order for printing, if it can first absorb the entire job file. In 1-to-N jobs, the value of N is usually not known prior to the start of job processing, in which case end-of-job must be programmed via a PspNextBankRequest toward the end of the job. In a 1-to-N job, image numbering corresponds directly to the logical page numbering of the document. Thus, the lowest image number represents the first logical page of the output document from the human reader's viewpoint. Note that though the images are processed in the general order 1-to-N (or N-to-1), they may be printed temporarily out of sequence, according to the requirements of the IOT to maintain maximum productivity with duplex paper paths and complex output collation devices. In any case, within a copy set, they are always in the proper sequence in the final output.

In N-to-1 printing jobs, the value of N is usually known prior to the start of job processing, so that end-of-job could be programmed via a PspNextBankRequest at the beginning of the job. In N-to-1 reprographic jobs, however, the value of N may not be known prior to the start of job processing, if the source is an unknown number of documents in the document handler of a scanner. For this reason, the image numbering of N-to-1 jobs must always be the inverse of the logical page numbering of the document, i.e., the lowest image number represents the last logical page of the output document from the human reader's viewpoint. End-of-job may be programmed via a PspNextBankRequest whenever N becomes known, which may be toward the end of reprographic jobs.

The sequence mode of each destination device is conveyed to the PSP in IotConfiguration. Note that some output devices may accept either sequence. In this case, the sequence is determined by the designation in the NextBankTaskInfoA parameter of PspNextBankRequest.

This numbering and sequencing convention accommodates IOTs with complex relationships between image scheduling and sheet scheduling, as well as those with simple relationships.

6.4.4 Page sync regimen

For purposes of this discussion, the term *page sync* will be taken to mean the page sync true interval which defines the slow scan dimension of the Standard Image Frame.

Each IOT must be designed so as to enable implementation of either of the following page sync regimens, according to the needs of the PSP with which it is to be associated.

a) Discrete regimen:

Page sync must occur at the physical interface for every real image. Page sync must not occur at the physical interface for dead cycles. Note, however, that during processing of a job, machine cycles which would otherwise be dead cycles may be utilized for on-line diagnostics during which a test image is sent to the IOT. Machine cycles which are utilized in this manner are considered to be *test cycles* rather than dead cycles, and are legitimately accompanied by page sync. Once the client layer imaging message sequence begins for a job, the sequence must continue for every page-time throughout the job, including dead cycles. Page sync must not occur at the physical interface at any time unrelated to the client layer message sequence, unless enabled during a NonProductive substate invoked by PspRequestIotDiagnostic and/or reported via IotStateInfo. It is shown in section 6.5 that during cycle-up, cycle-down, and dead cycles which occur while printing, imaging messages occur in the absence of page sync at the physical interface. But these messages must be related, as prescribed in section 6.4.2, to the stabilized page-time sequence internal to the IOT, so that they bear a known timing relationship to preceding and/or following page syncs.

b) Continuous regimen:

Page sync must begin as soon as possible after cycle-up, but no later than one page time prior to delivery of the first image, i.e., concurrently with the first IotVideoRequest status message, and must occur concurrently with every subsequent IotVideoHint status message regardless of whether or not the same page sync interval contains an IotVideoRequest message, or whether or not the related IotVideoRequest message is for a dead cycle. Once the client layer imaging message sequence begins for a job, the sequence must continue for every page-time throughout the job, including dead cycles.

6.4.5 Client layer intialization (refer to figure 6-6)

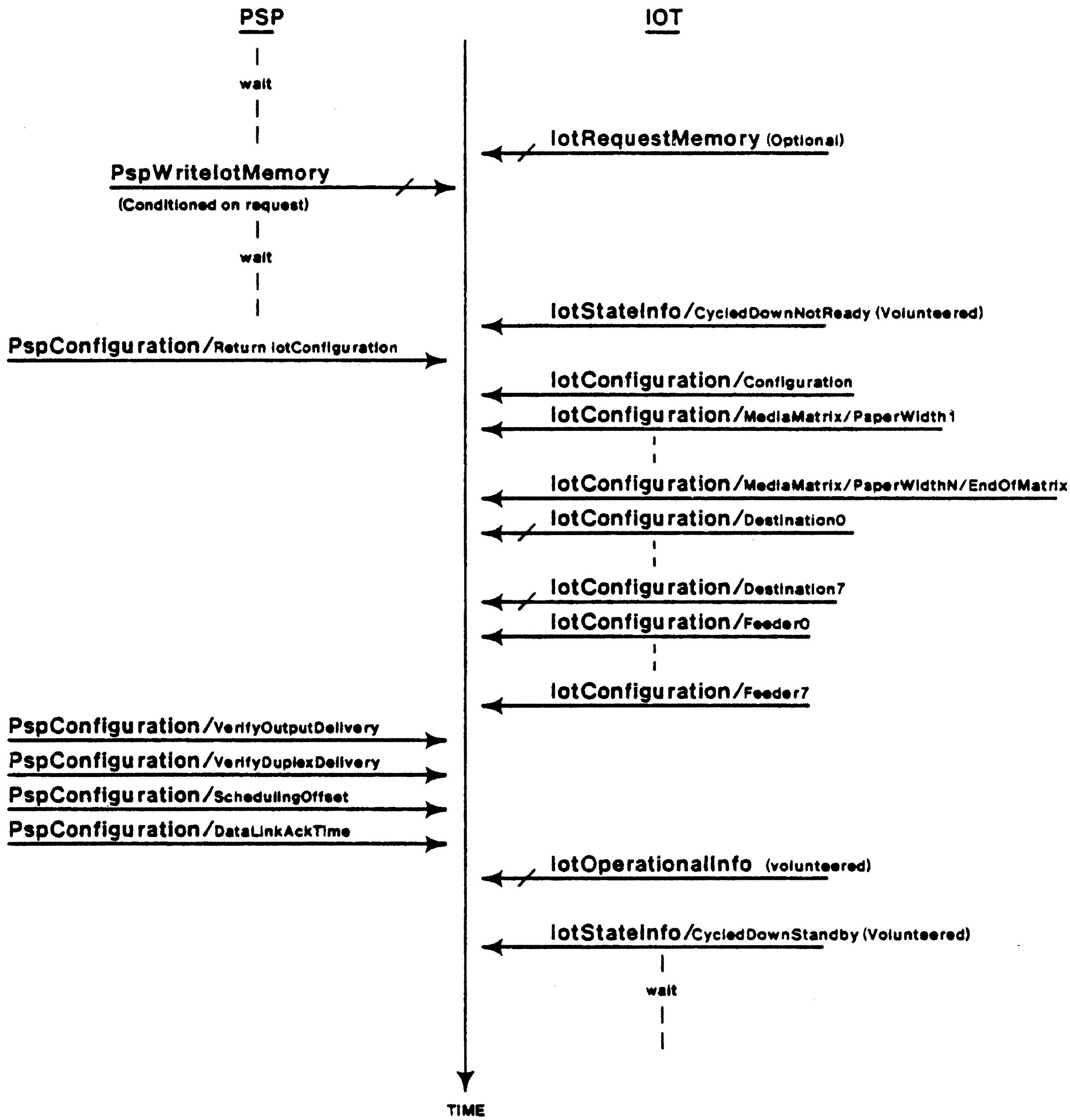
Assume that the Data Link has been established, as outlined in section 5.4. The PSP waits for the IOT to initiate the Client Layer communications (this wait should be timed out from the end of Data Link initialization). The IOT may begin with IotRequestMemory, if the IOT's software must be downloaded. The PSP response is PspWriteIotMemory. There may be multiple requests from the IOT, with corresponding responses from the PSP. (If there is no request from the IOT, the PSP does not write IOT memory.) When the IOT has all of its required memory, it volunteers IotStateInfo (probably CycledDownNotReady at this point) and waits for PspConfiguration. The first PspConfiguration message requests ReturnIotConfiguration. This request is sent first because the PSP needs the IOT Duplex Offset information in order to formulate its own SchedulingOffset requirement. The IOT responds accordingly with the full sequence of messages which constitute all of the IotConfiguration information. (The end of the sequence can be determined by the PSP from the contents of the messages. The number of InformationTypes is defined herein, and the number of entries under each type is either defined, or, the last entry is identified.) The PSP follows with its configuration information in a series of four PspConfiguration messages. The IOT next volunteers the full complement of operational information via multiple IotOperationalInfo messages. The PSP waits, and, when ready, the IOT volunteers IotStateInfo-CycledDownStandby. The printer is now ready to cycle-up and print.

6.4.6 Job interrupt protocol

Job interrupts which are to occur on a *set-boundary* (of a collated job) or a *stack-boundary* (of an uncollated job), will be handled by the PSP. The PSP will issue a PspSheetBankAbort specifying JobAbortWithRecovery and specifying a SheetNumber/CopyNumber that corresponds to the first sheet of the next set, or stack, of the job. This will effectively terminate the current job at the next set, or stack, boundary. The PSP will then program the interrupting job as any other job, i.e., it will be assigned the next unused JobNumber in the sequence, and the IOT will not be aware that a job interrupt has occurred. It is the PSP's responsibility to re-program the remaining sets, or stacks, of the interrupted job, sometime after completion of the interrupting job. The reprogrammed part of the interrupted job is assigned the next JobNumber in sequence at the time of re-programming. (This may or may not be next in sequence to the interrupting JobNumber, depending on whether or not the interrupted job is re-programmed immediately, and whether or not any next jobs are queued.) Thus, the re-programmed portion of the interrupted job also appears to the IOT as any other job. Note that this protocol supports nested set-boundary and/or stack-boundary interrupts solely under control of the PSP, up to the limit of the PSP to manage the job queue.

Job interrupts which must occur *as-soon-as-possible* (ASAP), will be handled jointly by the PSP and the IOT. The probability is that the interrupt will occur on a page-boundary of a collated set, or between copies in an uncollated stack. The interrupting jobs are assumed to be priority jobs, which must be scheduled by the IOT at the earliest opportunity. The PSP must program the interrupting job via a PspNextBankRequest, specifying InterruptJobAsap, specifying offset or, preferably, a different destination than the interrupted job, specifying a SheetNumber next in sequence to the last SheetNumber hinted by the IOT, and specifying the next unused JobNumber in the sequence. At its first opportunity, the IOT must issue an IotVideoHint specifying the first SheetNumber/CopyNumber and JobNumber of the interrupting job. If the IOT cannot effect the interrupt within one page-time, the first SheetNumber(s) of the interrupting job will duplicate the last SheetNumber(s) of the interrupted job; however, the JobNumbers remove any ambiguity. The last sheet of the interrupting job will be programmed as usual, i.e., PspNextBankRequest specifying EndOfJob and the appropriate SheetNumber with the interrupting JobNumber. The PSP will tell the IOT to resume the interrupted job by programming a PspNextBankRequest specifying ResumeInterruptedJob and specifying the JobNumber of the interrupted job. (All other parameters of this message are to be ignored by the IOT.) The IOT is responsible for logging the parameters of the interrupt point, for maintaining all banks of the interrupted job during the interrupt, and for resuming the interrupted job with an IotVideoHint for the next SheetNumber/CopyNumber following that of the last sheet which was printed and delivered to the output destination before the interrupt took effect. (An ASAP interrupt of a collated job is illustrated in section 6.5.) Note that this protocol supports nested ASAP interrupts up to the capacity of the IOT to store the context and banks of interrupted jobs, or as limited by the number of different destinations available.

Figure 6-6. **Client layer initialization**



6.4.7 Crash recovery protocol

If the IOT crashes or stops operating for any reason, it attempts to send an IotMetaReset/SelfReset message to the PSP. The accompanying Data Link procedure depends on whether Data Link communication is possible. If it is, IotMetaReset/SelfReset is sent successfully and the IOT performs the reset. After receiving IotMetaReset/SelfReset, the PSP Client Layer should request its Data Link to send the Disconnect command, and the IOT must then place its Data Link entity in the Asynchronous Disconnect Mode (ADM) according to the procedures in paragraph 5.4.4. (If Disconnect is not received from the PSP, the IOT Data Link should send Request Disconnect to elicit the Disconnect command from the PSP.) If Data Link communication is not possible, IotMetaReset/SelfReset transmission will fail. The IOT must then proceed with the reset and unilaterally place its Data Link entity in ADM. The PSP, having either sent the Disconnect command or detected the loss of Data Link communication, should periodically have its Client Layer request its Data Link to attempt to re-establish Data Link communications according to the procedures in section 5.4. When Data Link communication has been restored, the Client Layer is re-initialized using the initialization procedure of section 6.4.5. Following Client Layer re-initialization, the PSP optionally requests crash recovery status from the IOT by sending PspReadIotOperational Info/CrashRecoveryStatus (to which the IOT responds by sending one or more IotOperationalInfo/CrashRecoveryStatus). Finally, the PSP initiates job recovery by sending PspRequestIotAction/CycleUp, and the IOT must then send IotVideoHint for the appropriate image. This can always be done, since the IOT must save its context and parameter banks in non-volatile memory prior to hinting images from those banks.

If the PSP crashes or stops operating for any reason, the PSP attempts to send a PspMetaReset command to the IOT. The accompanying Data Link procedure depends on whether or not Data Link communication is possible. If it is, PspMetaReset is received successfully and the IOT must then perform the reset. After successfully sending PspMetaReset, the PSP Client Layer should request its Data Link to send the Disconnect command. The IOT must then place its Data Link entity in the Asynchronous Disconnect Mode (ADM) according to the procedures in paragraph 5.4.4. (If Disconnect is not received from the PSP, the IOT Data Link should send Request Disconnect to elicit the Disconnect command from the PSP.) If Data Link communication is not possible, PspMetaReset will not be received. When the IOT detects loss of Data Link communications, it must perform a reset and unilaterally place its Data Link entity in ADM. In either case, the IOT must take action to empty the paper path and cycle down, and to preserve its context and parameter banks. If the PSP crashes, the IOT must not introduce any spurious paper into the paper path nor produce any spurious output. Upon reload of the PSP software, the PSP attempts to re-establish Data Link communication according to the procedures in section 5.4. After Data Link communication is established, the IOT re-initializes the Client Layer as described in section 6.4.5. The PSP optionally requests crash recovery status from the IOT by sending PspReadIotOperationalInfo/CrashRecoveryStatus (to which the IOT responds by sending one or more IotOperationalInfo/CrashRecoveryStatus). Finally, the PSP initiates job recovery by sending PspRequestIotAction/CycleUp, and the IOT must then send IotVideoHint for the appropriate image.

Although the PSP periodically saves its context, it must depend on the IOT to recover jobs, since only the IOT can know with certainty the state of the paper path following the PSP's crash.

6.4.8 Fault protocol

Only those IOT conditions which warrant an IOT state change or substate change, or which warrant preventing an IOT state change or substate change, are to be considered faults. The PSP must be given notice of the occurance of any detectable fault or faults via an IotStateInfo message, and the identity of the fault or faults must be conveyed to the PSP as FaultCode(s) in one or more IotOperationalInfo messages. When multiple faults occur, either simultaneously or sequentially, IotStateInfo need be sent only once to notify of the initial state and/or substate change.

Those detectable IOT conditions which do not prevent the current job from being processed, but which represent an impending fault condition, should be classified as fault warnings and sent initially under the HintMessage InformationType of IotOperationalInfo. They should be elevated to detected faults when a threshold of the monitored parameter and/or other appropriate parameter is exceeded. (The appropriate parameter may be, for example, time.) Those faults detected in a device or facility of the IOT not employed in the current job (and which do not pose a hazard to machine or operator), should also be classified as fault warnings and reported via HintMessage. If and when said device or facility is selected for use, a fault should be reported, as described above.

It is desirable that correction of the physical condition which has been detected as a fault, should clear the related fault indication from the IOTs fault handler. If there are no remaining faults, this will cause a substate change (from FaultDetected to FaultNotDetected), which must be reported via an IotStateInfo message and an updated IotOperationalInfoMessage. If there are remaining faults, there will be no substate change, but an IotOperationalInfo message must still be sent to indicate the updated list of remaining faults.

It is recognized that clearing a fault indication as a result of correcting the physical condition may not be feasible, when detection of a fault causes transition to a state in which the monitored condition ceases to exist. (In such a case, correction of the fault condition could be verified only by returning, or attempting to return, to the state in which the fault was initially detected.) When such faults occur, the IOT should automatically clear the fault indication from its fault handler immediately after it is reported across the interface. Thus, the PSP will be notified of the related IOT state or substate change and its cause, but will be left unincumbered by a persisting fault report. The PSP will, therefore, be free to attempt to proceed, following the next appropriate operator action. If the fault condition persists, it will again be detected, again cause the state change, and again be reported across the interface. It is suggested that when a fault of this class is reported and then cleared as described above, the User Interface should maintain display of the fault, but advise the operator that it is permissible to attempt to resume operations. Resuming should cause the fault display to disappear. It would subsequently reappear or not, depending on whether the fault condition persisted or not. This procedure is intended to provide an acceptable degree of conformity with the philosophy that fault conditions should be perceived to be corrected by operator action.

Note: In the foregoing, the term *operator* is used in the broad sense of anyone operating the system.

6.5 Operating sequences, illustrated examples

The purpose of the following discussion is to illustrate typical operational sequences involving Client Layer messages and interface signals. The *discrete* page sync regimen described in paragraph 6.4.4 is assumed throughout. A large number of different sequences is possible. The differences depend on whether the job is simplex only, duplex only, or combination simplex-duplex. The differences will also be IOT specific, depending on the scheduling offset (defined above), the length of the duplex paper path (if present), whether or not the duplex path temporarily stores sheets, whether or not the output is to be collated, and on the requirements of the finishing station. The following examples assume processing of a single collated copy to a simple stacker, Scheduling Offset = 1, and a storing duplex paper path of length 9, i.e., minimum Duplex Offset = 9. Appendix B shows the equivalent set of sequences when the Scheduling Offset is 3. Appendix C shows processing of multiple collated copies to a bindexer output. The imaging messages, IotVideoHint, PspPrint, and IotVideoRequest, contain four numbering parameters: PlateNumber, SheetNumber, CopyNumber, and JobNumber. For simplicity, the accompanying figures omit those numbers which are not pertinent to the example.

6.5.1 Cycle-up sequence

Figure 6-7-a illustrates a simplex cycle-up sequence. It is assumed that both the Data link and the Client Layer have been initialized, as described in sections 5.4.1, 5.4.2, and 6.4.5. The PSP first sends one or more PspNextBankRequests to program the job. This is followed by PspRequestIotStateChange-CycleUp. When appropriate, the IOT volunteers IotStateInfo-CycledUpPrinting-TaskInProcess and begins the sequence of imaging messages. Neither client layer imaging messages nor **Page sync** may appear at the interface until page-time is stabilized to the specifications of the IOT. The messages will precede page sync because of the scheduling offset, the necessity for Request to precede the page sync interval during which the image is delivered, and the restriction that page sync may be present only when the video for an actual image is being delivered. The imaging message sequence begins with the Hint, Print, and Request triplet for the first image, and continues with the triplets for the second and subsequent images. The first transition of page sync occurs in the page-time following the Request for the first image. Note that all Hint, Print, and Request messages must conform to the timing requirements given in figure 6-4, including those which precede the onset of **page sync**. The latter must be timed with reference to the IOT internal slow scan time base. The image generator must rely on the Hint preceding the first page sync transition as the timing reference for delivery of video for the first image. Subsequently, the image generator can use the previous page sync transition, which is a more accurate reference. The number of messages which occur prior to page sync will depend on the scheduling offset. Figure 6-7a shows the minimum. For example, refer to figure B6-7a in appendix B for a case with scheduling offset of 3. A similar sequence for scheduling offset = 1 and two copies to a bindexer output destination is shown in figure C6-7-a of appendix C.

6.5.2 Cycle-down sequence

Figure 6-7-b illustrates the simplex cycle-down sequence. It is assumed that the IOT has been cycled-up and printing. The last image to be printed from the current bank is image-n. It is assumed that the value of (n) was not known beforehand; therefore, cycle-down (End-of-job) is programmed with a PspNextBankRequest near the end of of the job. In figure 6-7-b, the PspNextBankRequest is shown occuring during the latest page-time permissible. This is followed by a Print dead cycle in the same page-time, and then a sequence of page times containing Request dead cycle, Hint dead cycle, and Print dead cycle. This sequence must continue at least one cycle beyond the page-time during which the last image is delivered, so that the final Hint, occurring during the first dead cycle, can provide a timing reference for re-imaging the last image, in the event that it is aborted. Note that the sequence would look slightly different if the PspNextBankRequest were issued during an earlier cycle. In that case, a Hint dead cycle would replace the Hint n+1. When the scheduling offset is greater than one, there is a corresponding number of additional dead cycles. Refer to figure B6-7-b, appendix B. A similar sequence for scheduling offset = 1 and two copies to a bindexer output destination is shown in figure C6-7-b of appendix C.

6.5.3 Duplex cycle-up sequence

A duplex cycle-up sequence is shown in figure 6-8. The sequence is initially the same as figure 6-7-a (the plate numbers indicating simplex side and duplex side are included in the illustration). The sequence continues with simplex side images for nine page times, corresponding to the assumed duplex paper path length, then alternates between duplex side and simplex side images. This is characteristic of duplex paper paths. However, some non-storing, or *racetrack*, duplex paper paths may require alternate dead cycles in the initial simplex sequence, so as to allow for subsequent interleaving of the simplex and duplex images. Recent racetrack designs do not require dead cycles. Note also that the DuplexOffset (refer to IotConfiguration) increases (from 9 to 17, in this example) as the duplex path is filled. This is characteristic of a storing duplex paper path. With a non-storing or racetrack duplex path, the DuplexOffset remains constant. A corresponding sequence for scheduling offset of 3 is shown in figure B6-8 of appendix B. A similar sequence for scheduling offset = 1, two copies to a bindexer output destination, and a racetrack duplex paper path with DuplexOffset = 8, is shown in figure C6-8 of appendix C.

6.5.4 Duplex cycle-down sequence

A duplex cycle-down sequence is shown in figure 6-9. The PspNextBankRequest for End-of-job at sheet 19 occurs during page sync for image 9,2, preceding Hint 19,1 and Print 19,1. The sequence, therefore, reverts to continuous duplex sides at the next opportunity, i.e., during page sync for image 19,1, the IOT begins hinting for image 12,2 (to follow 11,2) rather than for 20,1. After the last valid image (19,2) is requested, the IOT begins to hint dead cycles. When the duplex paper path is depleted, the sequence continues so as to provide the final Hint, as in figure 6-7-b. The corresponding sequence for scheduling offset of 3 is shown in figure B6-9, appendix B. A similar sequence for a scheduling offset = 1, two copies to a bindexer output destination, and a racetrack duplex paper path with DuplexOffset = 8, is shown in figure C6-9 of appendix C.

6.5.5 Duplex-to-simplex transition

Figure 6-10 illustrates a duplex to simplex transition. The sequence is similar to the duplex cycle-down case, except that after it depletes the duplex paper path with an all duplex-side sequence, it continues with an all simplex-side sequence.

6.5.6 Simplex abort sequence

A simplex abort sequence is shown in figure 6-11. A PspSheetBankAbort of sheet 10 is assumed to occur during page-time for image 10. This image will, therefore, be discarded (this may result in a hole in the paper path, or in a discarded sheet bearing the damaged image, depending on when paper is committed in a particular IOT). Since image 11 has already been requested, it will be imaged and will also have to be discarded. The PspSheetBankAbort intervenes between Hint 12 and the expected Print 12. Hint 12 is, therefore, answered with Print-dead-cycle to allow the IOT to recover. Following the Request-dead-cycle response, the Hint, Print, Request sequence has recovered, and the imaging sequence recovers following the dead cycle. Note that page sync does not occur during the dead cycle, and the image generator in the PSP must rely on H1,1 for its timing reference for reimaging image 10. The corresponding sequence for scheduling offset of 3 is shown in figure B6-11 of appendix B. In this case three dead cycles are required to effect recovery.

Note: The number of dead cycles required for recovery is dependent on the scheduling offset. It may also be job dependent and IOT dependent in other respects. The number shown is the minimum for the conditions assumed.

6.5.7 Duplex abort sequence (simplex side)

Figure 6-12 illustrates a duplex abort sequence in which the sheet abort occurs during imaging of simplex side 15. Note that duplex side imaging continues uninterrupted, while simplex side imaging recovers in an interleaved fashion. The aborted image is the only one lost, and only one dead cycle is required to recover. The corresponding sequence for scheduling offset of 3 is shown in figure B6-12 of appendix B. In this case, two dead cycles are required to effect recovery (refer to note in section 6.5.6).

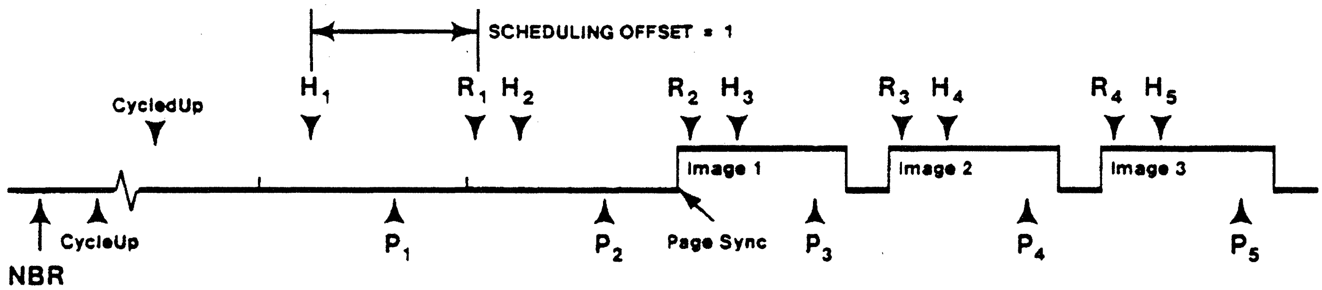
6.5.8 Duplex abort sequence (duplex side)

Figure 6-13 illustrates a duplex abort sequence in which the sheet abort occurs during imaging of duplex side 4. In this case, recovery must start with simplex side 4. Therefore, the original sheet 4 is lost, the next in sequence simplex image, 13, is lost (since it has already been requested and would now be out of sequence), and the duplex path is flushed of all its simplex imaged sheets (5 through 12). This is followed by an all simplex sequence which refills the duplex path with re-imaged simplex sides 4 through 12. Sheet 4 is now available to accept the originally aborted duplex side 4, and the normal simplex-duplex alternating sequence resumes. The corresponding sequence for scheduling offset of 3 is shown in figure B-6-13 of appendix B. In this case, recovery is achieved with the same number of dead cycles (refer to note in section 6.5.6).

6.5.9 ASAP interrupt sequence (simplex jobs)

Figure 6-14 illustrates an ASAP interrupt sequence in which a simplex job, JobNumber 7, is interrupted by a five-page simplex job, JobNumber 8 (for simplicity, PlateNumbers and CopyNumbers are omitted from the figure). In this case, the interrupt is initiated by a PspNextBankRequest following the Hint for SheetNumber 11, JobNumber 7, and designates the first sheet of the interrupting job as number 12. The IOT, however, is unable to effect the interrupt immediately, and does not begin to Hint SheetNumber 12, JobNumber 8 until SheetNumber 12, JobNumber 7 is being imaged. This results in reusing SheetNumbers 12 and 13 in the interrupting job, but the JobNumbers preclude ambiguity. The interrupting job is ended normally by means of a PspNextBankRequest specifying EndOfJob at SheetNumber 16, JobNumber 8. A PspNextBankRequest specifying ResumeInterruptedJob, JobNumber 7, causes the IOT to return to the job #7 parameter banks and resume Hinting with the SheetNumber next in sequence following the interrupt point. The IOT is responsible for chosing the actual interrupt point in the sheet sequence of JobNumber 7, and for saving the context and parameter banks of the interrupted job (also refer to section 6.4.6).

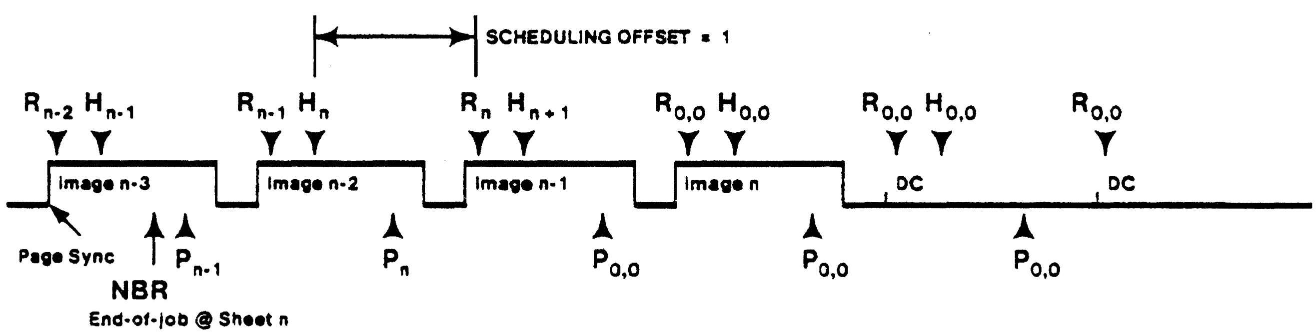
Figure 6-7-a. **Simplex cycle–up sequence**
(scheduling offset = 1)



$R_{0,0}$ ::= Request for a Dead Cycle (DC)

H_n P_n R_n ::= n-th "Hint, Print, Request" triplet. (Only sheet nos. are shown.)

Figure 6-7-b. **Simplex cycle–down sequence**
(scheduling offset = 1)



$H_{0,0}$ ::= "Hint" Dead Cycle (DC)

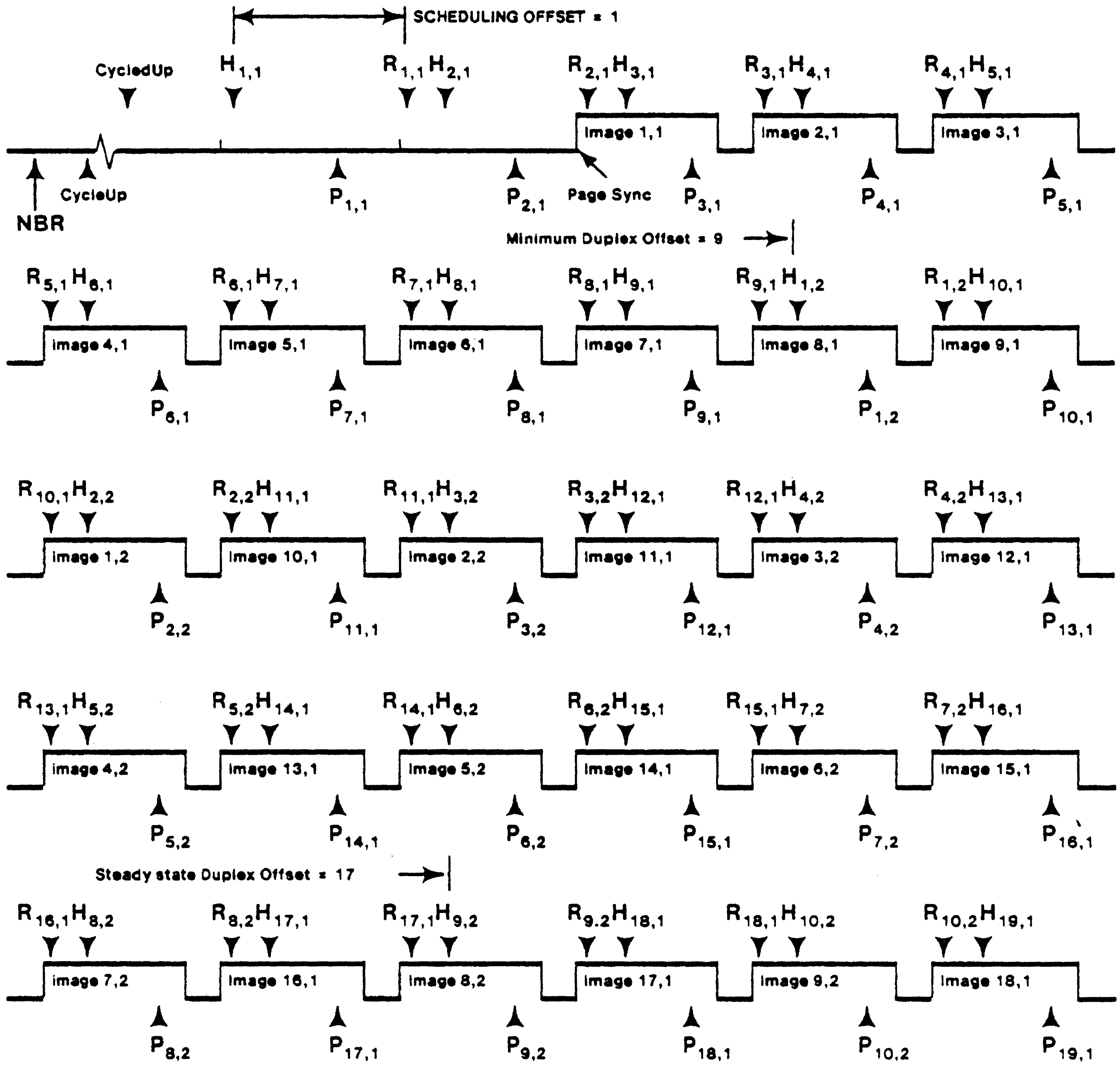
$P_{0,0}$ ::= Print Dead Cycle

$R_{0,0}$ ::= Request for a Dead Cycle

H_n P_n R_n ::= n-th "Hint, Print, Request" triplet. (Only sheet nos. are shown.)

NBR ::= NextBankRequest

Figure 6-8. **Duplex cycle-up sequence**
(scheduling offset = 1)
(Storing duplex paper path, minimum duplex offset = 9)



$H_{i,j}$ $P_{i,j}$ $R_{i,j}$ ::= "Hint, Print, Request" triplet for sheet i, plate j.

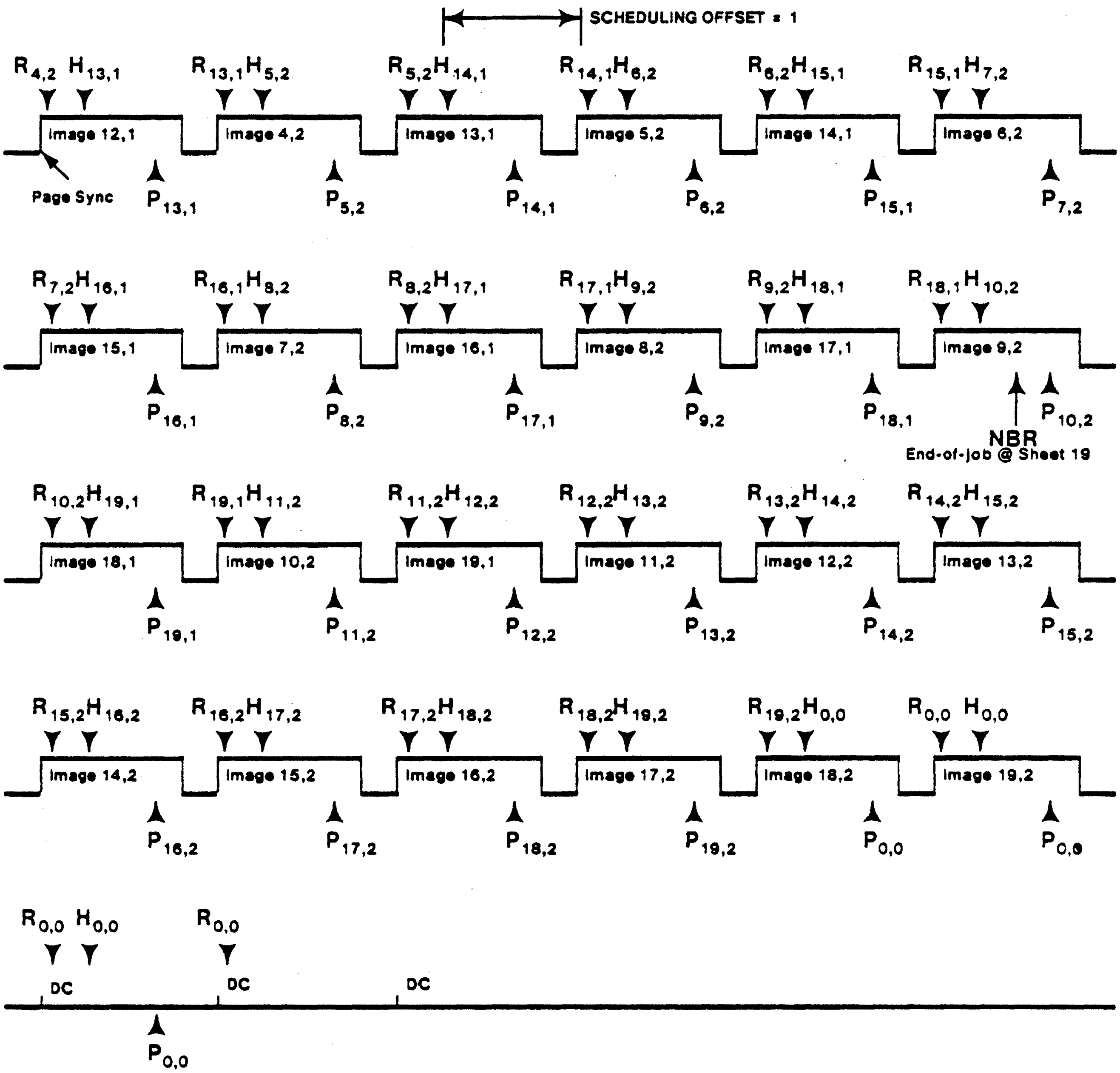
j ::= PlateNumber

PlateNumber ::= simplex side | duplex side

simplex side ::= 1

duplex side ::= 2

Figure 6-9. **Duplex cycle-down sequence**
(scheduling offset = 1)
(Storing duplex paper path, minimum duplex offset = 9)



$H_{i,j}\ P_{i,j}\ R_{i,j}$ ::= "Hint, Print, Request" triplet for sheet i, plate j.

j ::= PlateNumber

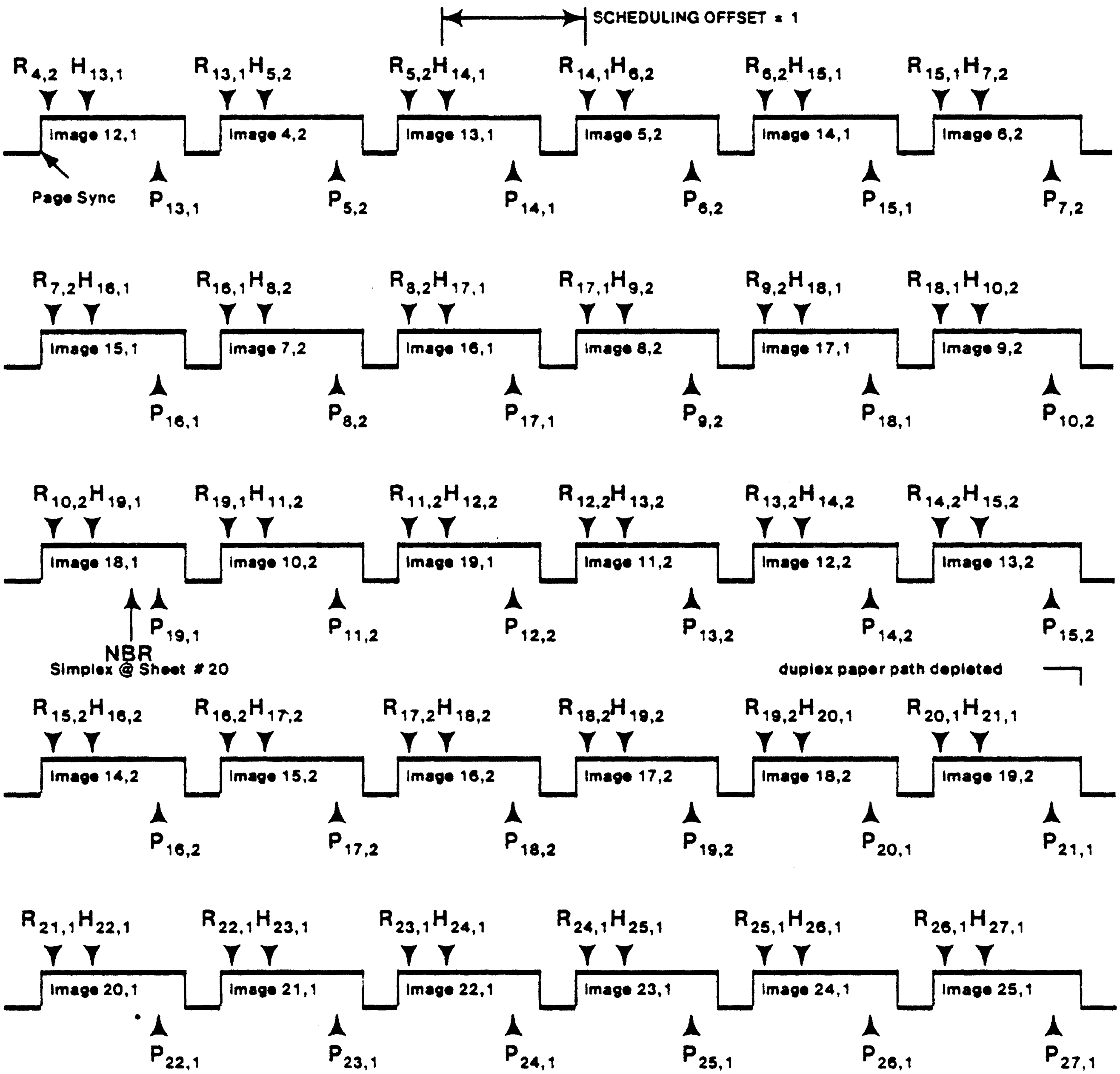
PlateNumber ::= simplex side | duplex side

simplex side ::= 1

duplex side ::= 2

NBR ::= NextBankRequest

Figure 6-10. **Duplex-to-simplex transition**
(scheduling offset = 1)
(Storing duplex paper path, minimum duplex offset = 9)



$H_{i,j}$ $P_{i,j}$ $R_{i,j}$ ::= "Hint, Print, Request" triplet for sheet i, plate j.

j ::= PlateNumber

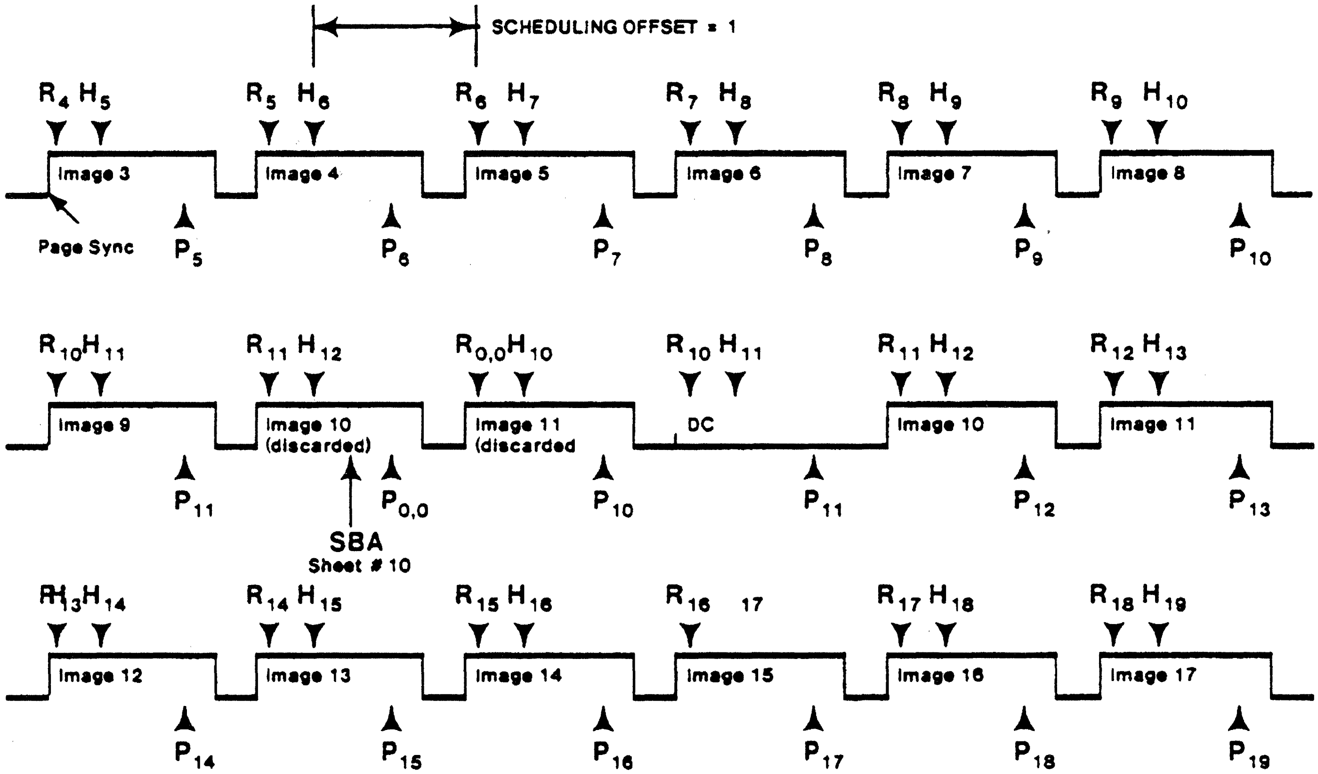
PlateNumber ::= simplex side | duplex side

simplex side ::= 1

duplex side ::= 2

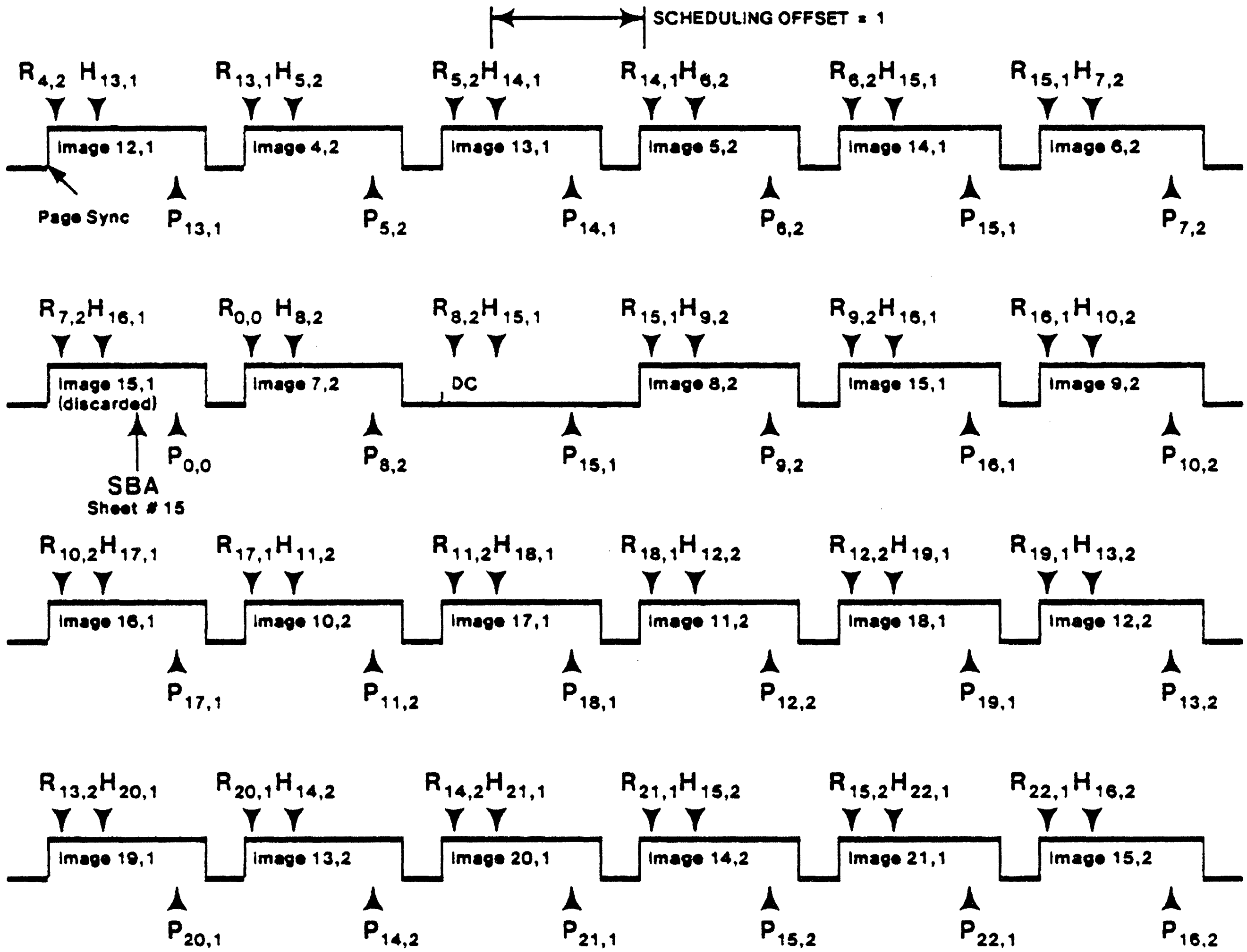
NBR ::= NextBankRequest

Figure 6-11. **Simplex abort sequence**
(scheduling offset = 1)



$H_{0,0}$	::=	"Hint" Dead Cycle (DC)
$P_{0,0}$	::=	Print Dead Cycle
$R_{0,0}$	::=	Request for a Dead Cycle
H_n P_n R_n	::=	n-th "Hint, Print, Request" triplet.
SBA	::=	SheetBankAbort

Figure 6-12. **Duplex abort sequence, sheet abort during simplex side imaging**
(scheduling offset = 1)
(Storing duplex paper path, minimum duplex offset = 9)



$H_{i,j}$ $P_{i,j}$ $R_{i,j}$ ::= "Hint, Print, Request" triplet for sheet i, plate j.

j ::= PlateNumber

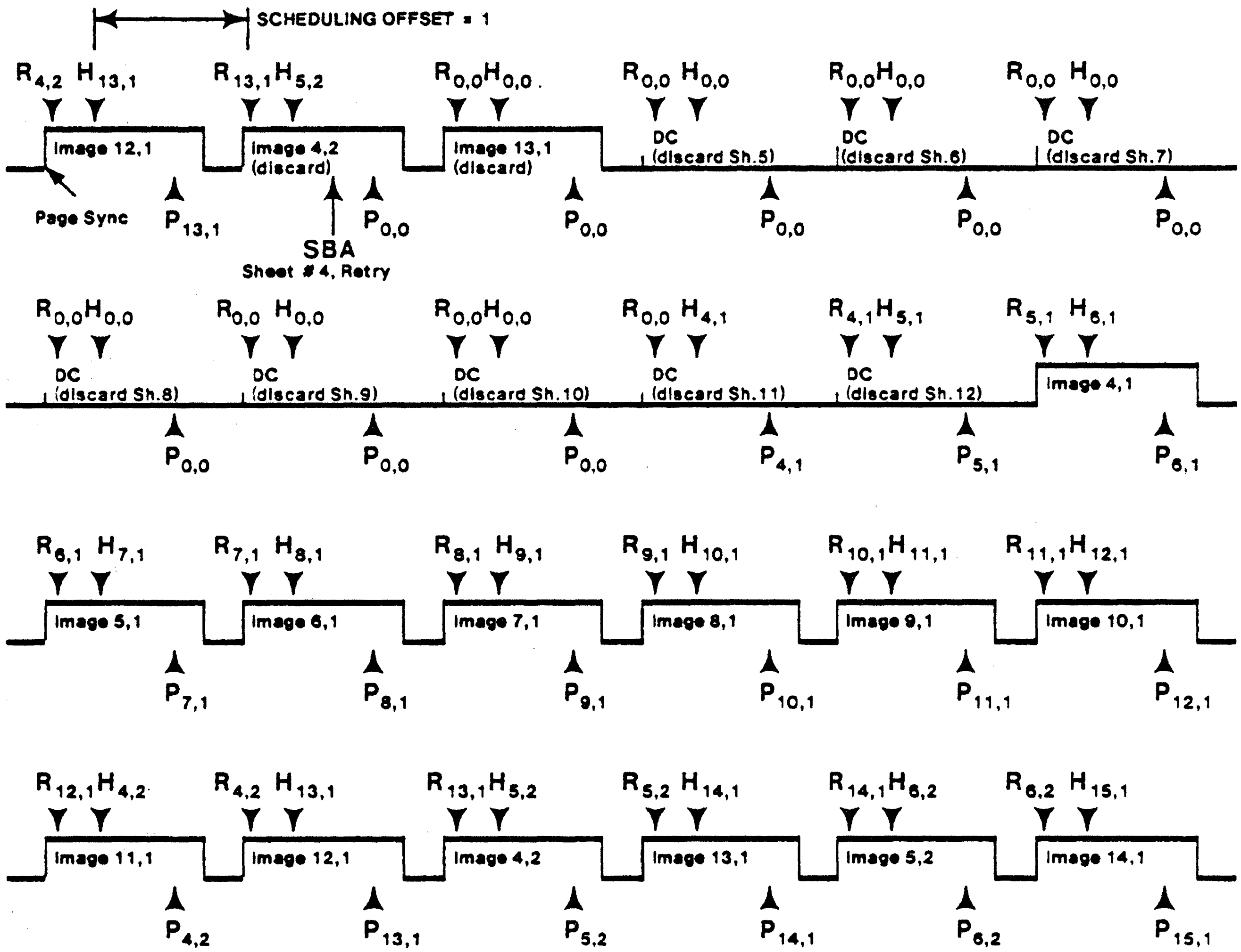
PlateNumber ::= simplex side | duplex side

simplex side ::= 1

duplex side ::= 2

SBA ::= SheetBankAbort

Figure 6-13. **Duplex abort sequence,**
sheet abort during duplex side imaging
(scheduling offset = 1)
(Storing duplex paper path, minimum duplex offset = 9)



$H_{i,j}\ P_{i,j}\ R_{i,j}$ ::= "Hint, Print, Request" triplet for sheet i, plate j.

j ::= PlateNumber

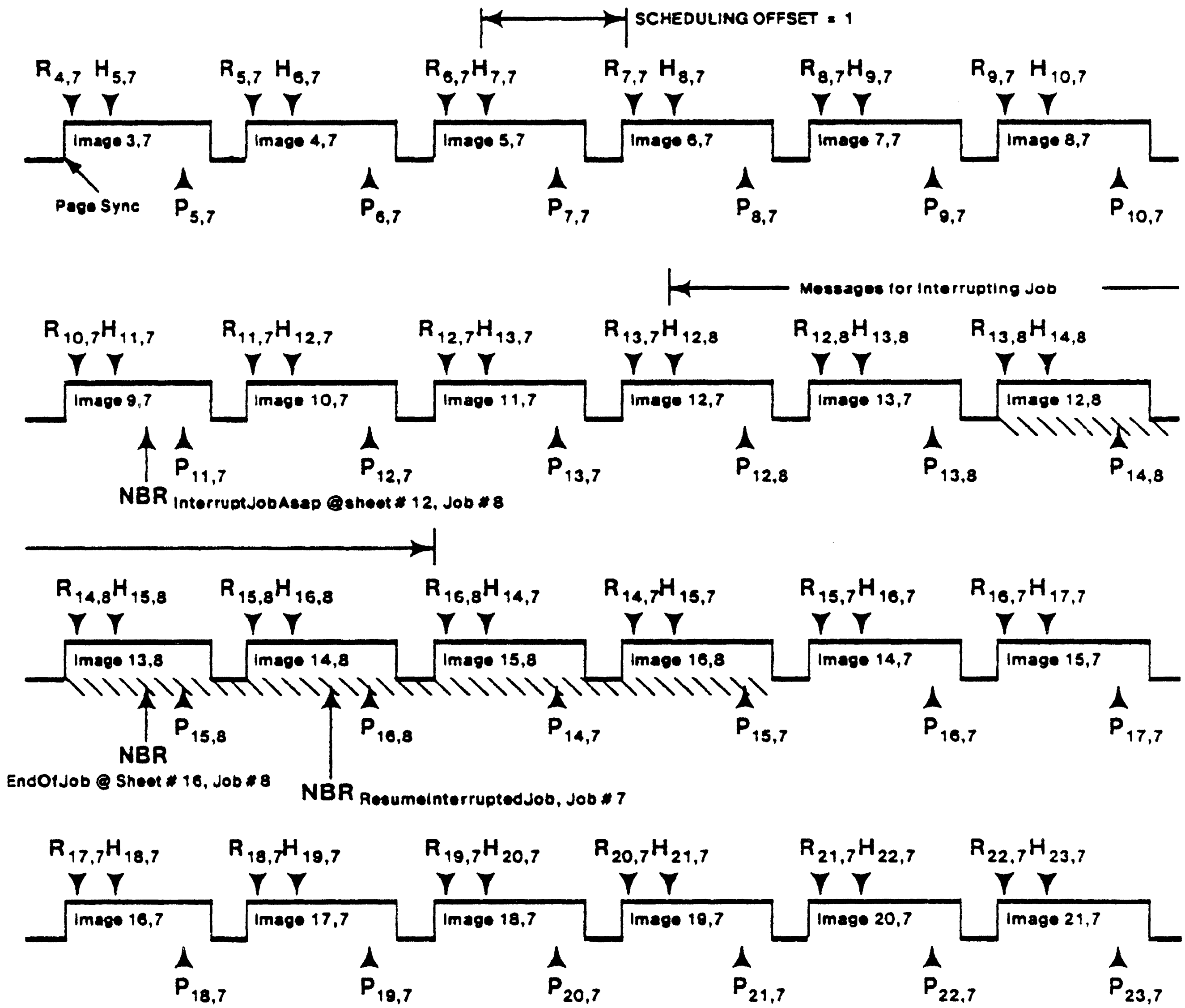
PlateNumber ::= simplex side | duplex side

simplex side ::= 1

duplex side ::= 2

SBA ::= SheetBankAbort

Figure 6-14. **ASAP interrupt, simplex jobs**
(Job #7 interrupted by a 5 page job, #8.)
(scheduling offset = 1)



$H_{i,j}\ P_{i,j}\ R_{i,j}$ ::= **"Hint, Print, Request" triplet for sheet i, job j (PlateNumber and CopyNumber not shown)**

NBR ::= NextBankRequest

6.6 IOT diagnostic protocol

No distinct diagnostic machine state or substate is prescribed herein. The diagnostic state may be considered to be a superset of the prescribed machine states and substates, i.e., diagnostics may be initiated in any machine state and substate.

There are two types of diagnostic utilities, active and passive. An active diagnostic utility renders the IOT unavailable for customer jobs in either the cycled-up or cycled-down machine state, and the accompanying Productivity substate must be *Service*. For example, an Active diagnostic utility may comprise one or more preprogrammed printing jobs. A passive diagnostic utility is a measuring or monitoring activity which can run concurrently with a printing job if necessary, without interrupting the job. If the Productivity substate is *Productive*, it may remain *Productive*. However, the IOT must report such passive diagnostic activity via the InfoMessage field of IotOperationalInfo, since no machine state or substate change occurs.

Diagnostic utilities are invoked by PspRequestIotDiagnostic. This command can be valid in any machine state. If the specific utility cannot be run because of priority activity in the IOT, the request must be denied via the *Reject* code in the Status field of IotDiagnosticResponse. A customer job, for example, must be considered priority activity. There may be other IOT-specific priority activities. The IOT indicates acceptance of PspRequestIotDiagnostic by echoing the contents of the Command field in the Status field of IotDiagnosticResponse. A sample diagnostic message exchange is illustrated in figure 6–15.

6.7 IOT configuration data

A great deal of IOT-specific information is conveyed to the PSP at start of operations, via the Status message, IotConfiguration. However, there remains some IOT-specific information which must be made known to the PSP by means other than this Generic Specification, or other than through operational exchange of Command/Status messages. This information is listed below:

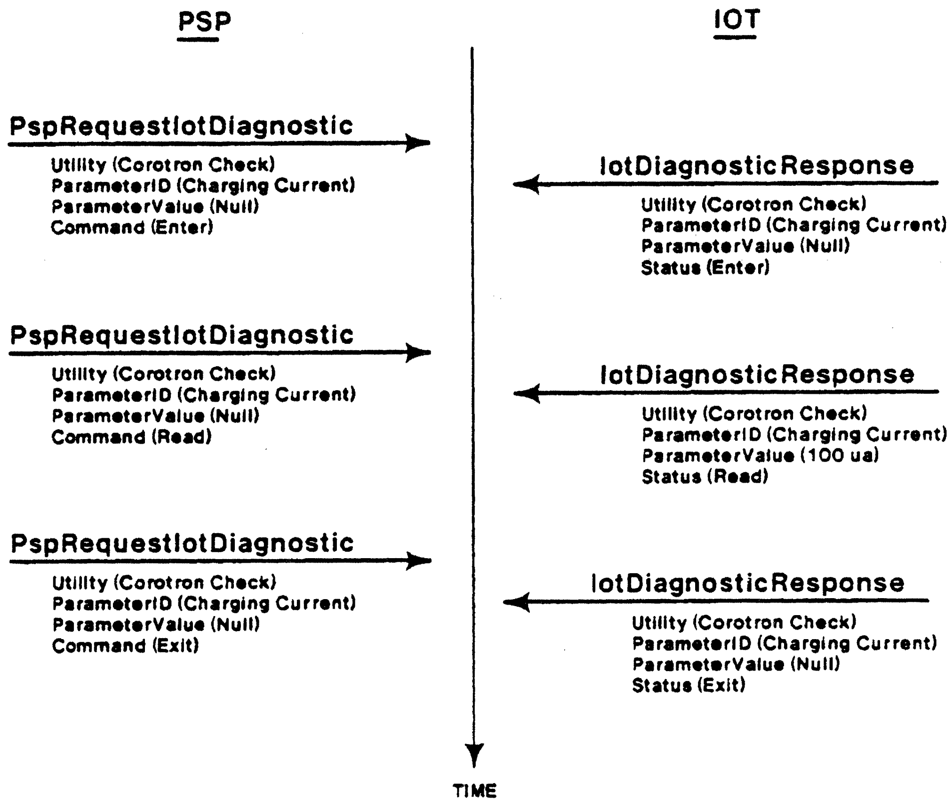
1. *Internal grounding arrangements:* The connection arrangements between signal ground and ground mecca within the IOT must be coordinated with the similar arrangements within the PSP, so as to achieve the desired noise and EME performance of the overall printer formed by interconnecting the particular PSP and the particular IOT with the cable prescribed herein (section 8.9).
2. *Coding of IOTMetaReset*, coarse codes and fine codes (section 6.3.3).
3. *Coding of IOT display message numbers*, to be used in PspRequestIotAction command (section 6.3.2).
4. *Fault codes and message codes*, to be used in IotOperationalInfo, for TechRepClearOnlyFaults, OperatorClearFaults, TechRepReTryFaults, HintMessage, and InfoMessage. Configuration type for IotConfiguration, fault numbers for TechRepReTryFaults (section 6.3.3).

5. *Coding of ConfigurationID,* to be used in status message, IotConfiguration (section 6.3.3).
6. *Diagnostic Utility and Utility Parameter descriptions and identification,* to be used in Command/Status messages, PspRequestIotDiagnostic and IotDiagnosticResponse (sections 6.3.2 and 6.3.3).
7. *Diagnostic Utility software,* which must be down loaded by the PSP to the IOT prior to running IOT diagnostics, unless this software is IOT resident.

6.8 PSP configuration data

There is no information about the PSP which must be made known to the IOT, beyond that which is prescribed in this Generic Specification or made known in the prescribed Command/Status message exchanges, except for the internal grounding arrangements implied in section 6.7.1.

Figure 6-15. **Sample diagnostic message exchange**



7. Video data signal quality

7.1 Transmission impairments

Transmission across the interface will subject video data and clock signals to propagation delay, to intersymbol interference, and to attenuation. Delay and intersymbol interference are caused by the active interface components, as well as by the cable. Attenuation is due only to the cable. Delay itself is not considered to be an impairment, but variations in absolute delay of the data relative to the clock can affect the ability to reclock the data without error. Gain in the line receiver compensates for attenuation, but if the loss of noise margin at the receiver input is severe, the amplified signal may contain errors caused by noise.

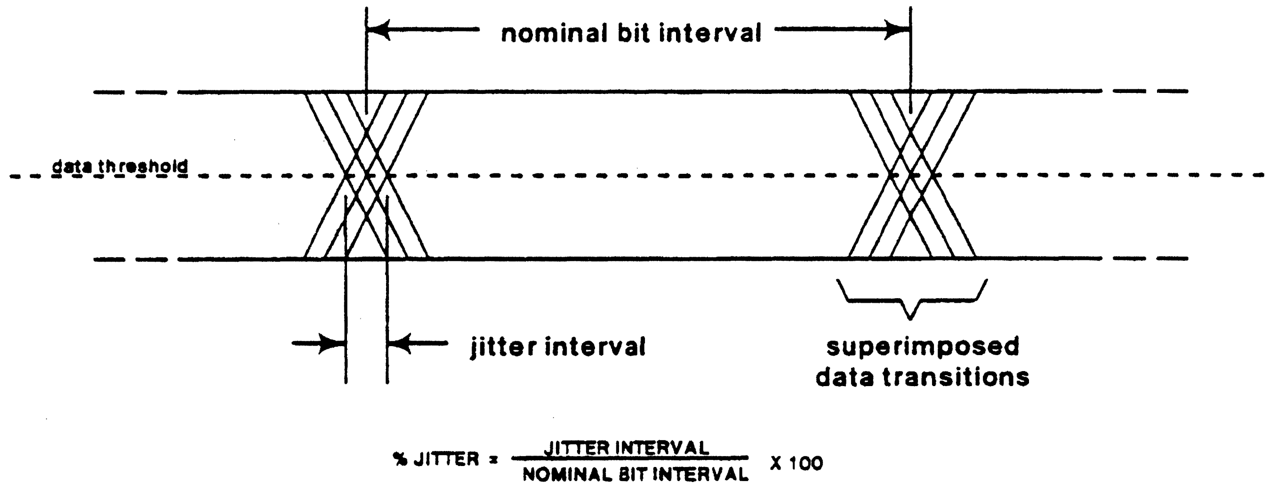
Intersymbol interference is caused by the spreading of the energy allocated to a bit interval into adjacent bit intervals. In general, the interference can spread over several bit intervals and can precede as well as trail the reference interval. With the recommended interface (chapter 8), the interference is almost all trailing. The interference is non-stationary, i.e., observable as time jitter when the data is non-stationary, i.e., non-repetitive. The interference is stationary with repetitive signals, such as a clock, and does not cause observable displacement of the clock transitions relative to each other. However, displacements can occur between clock transitions at the beginning or end of clock bursts, relative to transitions away from these extremes. When the strings of zeroes in the data, and the quiescent intervals of clock, both exceed the range of the intersymbol interference, the relative displacement of clock transitions is of the same order of magnitude as the jitter observed on the data. This is the case for the prescribed interface. Therefore, for purposes of analysis, the same amount of jitter will be assigned to both pixel clock and video data.

Jitter is quantified as the difference between the worst-case transition displacement in the negative time direction and the worst-case transition displacement in the positive time direction (i.e., peak-to-peak relative to the nominal position of the transition), and is stated as a percentage of the nominal pixel interval. These measurement relationships are shown in the stylized sketch in figure 7-1. The standard interface has been designed to provide less than 5 percent jitter at the output of the line receivers at 25 Megabits per second.

Cable attenuation results in loss of noise margin at the receiving end of the cable. The differential noise margin is defined as the difference between the magnitude of the differential signal at the line receiver input, and the minimum allowable differential input. The acceptable limits for ECL output logic levels are VOHAmin and VOLAmax. Thus, the minimum acceptable differential output is as follows:

[VOHAmin - VOLAmax] = [– 0.98v – (– 1.63v)] = 650 mv. The minimum allowable differential input is taken to be 200 mv. Therefore, the differential noise margin available for attribution to the cable is 450 mv. Measurements have shown that cable attenuation will be somewhat less than 3 db at 25 Mbps with 50 feet of cable. Again, assuming minimum ECL output of 650 mv from the line driver, the minimum differential signal arriving at the input to the line receiver with 3 db of attenuation would be 460 mv, and the realized differential noise margin would be 460 − 200 = 260 mv. (The normal ECL logic noise margins are 125 mv and 155 mv for the high and low logic levels, respectively.) Of course, the signal will also enjoy the common mode rejection of the recommended line receiver, +1 volt. In view of the above, cable attenuation is not considered to be a problem.

Figure 7-1. **Definition of video data jitter**



7.2 Timing analysis for reclocked modes

An example of the overall timing relationships for the reclocked modes is shown at the top of figure 7-2. The propagation delay due to the cable and components will cause a slight image shift in the fast scan direction, but this shift will be compensated by the image alignment process. Only signal degradations which cause reclocking errors will affect the quality of the video data delivered to the marking engine. The principle concern, therefore, is the reliability of reclocking, or more specifically, the phase relationship between video data and return clock, at the reclocking stage in the IOT interface. The analysis presented below applies equally to reclocking in the serial or parallel video data transmission mode, since worst case conditions will be considered.

It is assumed (but not required) that the proper phase relationship between **video data** and **return** (**pixel** or **byte**) **clock** is established at the PSP by generating the return clock as the inverse of the clock received from the IOT. This establishes the rising edge of the return clock nominally at the center of the video data interval. The steady state phase relationship between video data and return clock as received at the IOT, thus, depends on the duty cycle symmetry of the (pixel or byte) clock, plus the propagation delay tolerances of all the active and passive circuit elements in each transmission channel between the clocking point in the PSP and the reclocking point in the IOT. The transient phase relationship between video data and return clock as received at the IOT depends on the jitter characteristics of all the active and passive circuit elements in each transmission channel between the clocking point in the PSP and the reclocking point in the IOT. It is assumed that the transient displacements will add to the steady state displacements in the worst combinations. The model for the following timing tolerance analysis is shown at the bottom of figure 7-2. It is based on the circuits recommended in chapter 8.

Taking the clock at point C in figure 7-2 as the absolute timing reference, the video data is subjected to the propagation delays of the ECL clocking stage, the cable, and the ECL line receiver before it arrives at point F, the input to the ECL reclocking stage in the IOT. The return clock is subjected to the propagation delays of the ECL line driver (which also effects logical inversion), the cable, and the ECL line receiver before arriving at point G, the other input to the ECL reclocking stage. In addition to delay, jitter can cause video data and clock transition displacements of ±2.5 percent. The ECL flip-flop recommended for clocking has a propagation delay range of 1.5 to 3.7 ns. For the recommended drivers and receivers, the range of propagation delay is 1.0 to 2.8 ns. (These figures are based on the assumption that all the components in the PSP are at the same temperature (+85 degrees C) and all the components in the IOT are at the same temperature (+85 degrees C). The individual circuit functions, however, are not assumed to be on the same substrate.) The maximum difference in delay between any two pairs of the recommended cable is specified as 2 percent of the total delay. This includes differences in velocity of propagation, as well as differences in equivalent length. The nominal velocity of propagation, v_p, is 78 percent. The nominal propagation delay in nanoseconds per foot of cable is $1.017/v_p = 1.30$ ns/ft. Consider a −20 percent deviation from nominal for vp. The worst case differential delay between two pairs of a fifty foot cable is then (1.30 x 1.2) ns/ft x .02 x 50 ft. = 1.56 ns. (In the analysis, this difference is split between the signals, 0.78 ns relative advance and 0.78 ns relative delay.) The clock phasing tolerance is specified as ±5 percent in sections 4.2.1 and 4.2.2. In this analysis, it is assumed to be due to inverting a received clock with ±5 percent duty cycle tolerance.

With the above parameters, and assuming 25 Mhz clock and 25 Megabits per second data on any given video data path, the earliest clock at point G in figure 7-2 will be delayed 1.0 ns each by the line driver and receiver, and advanced 0.78 ns by the cable, and 2.0 ns by −5 percent initial phasing error and 1.0 ns by −2.5 percent jitter, for a relative advance of 1.78 ns. The latest clock will be delayed 2.8 ns each by the line driver and receiver, 0.78 ns by the cable, 2.0 ns by +5 percent initial phasing error, and 1.0 ns by +2.5 percent jitter, for a relative delay of 9.38 ns. The earliest data at point F in figure 7.2 will be delayed 1.5 ns by the clocking stage in the PSP, 1.0 ns by the line receiver, and advanced 0.78 ns by the cable, and 1.0 ns by −2.5 percent jitter, for a relative delay of 0.72 ns. The latest data will be delayed 3.7 ns by the clocking stage in the PSP, 2.8 ns by the line receiver, 0.78 ns by the cable, and 1.0 ns by +2.5 percent jitter, for a relative delay of 8.28 ns.

These figures are summarized in figure 7-3, which shows that at 25 Megabits per second and with 50 feet of cable, there is a margin of 9.94 ns between earliest clock and latest data, and a margin of 11.34 ns between latest clock and earliest data. These margins are adequate for reliable reclocking, *provided that the* *first stage of reclocking following the cable receivers is implemented with ECL components.*

If TTL were to be used for the clocking and reclocking stages, TTL/ECL and ECL/TTL translators would have to be used for the line drivers and receivers, respectively, and an extra translator would be needed at the output of the clocking stage in the PSP. The set-up time for TTL reclocking would be somewhat longer than for ECL, and the translators have a wider range of propagation delay than the ECL devices. A careful analysis should be carried out for any TTL design. It is estimated that the maximum data rate would be around 12.5 Megabits per second.

Figure 7-2. **Overall timing relationships for reclocked video data modes**
(Assuming 25 Mbps, and 50 ft of cable @ 1.3ns/ft)

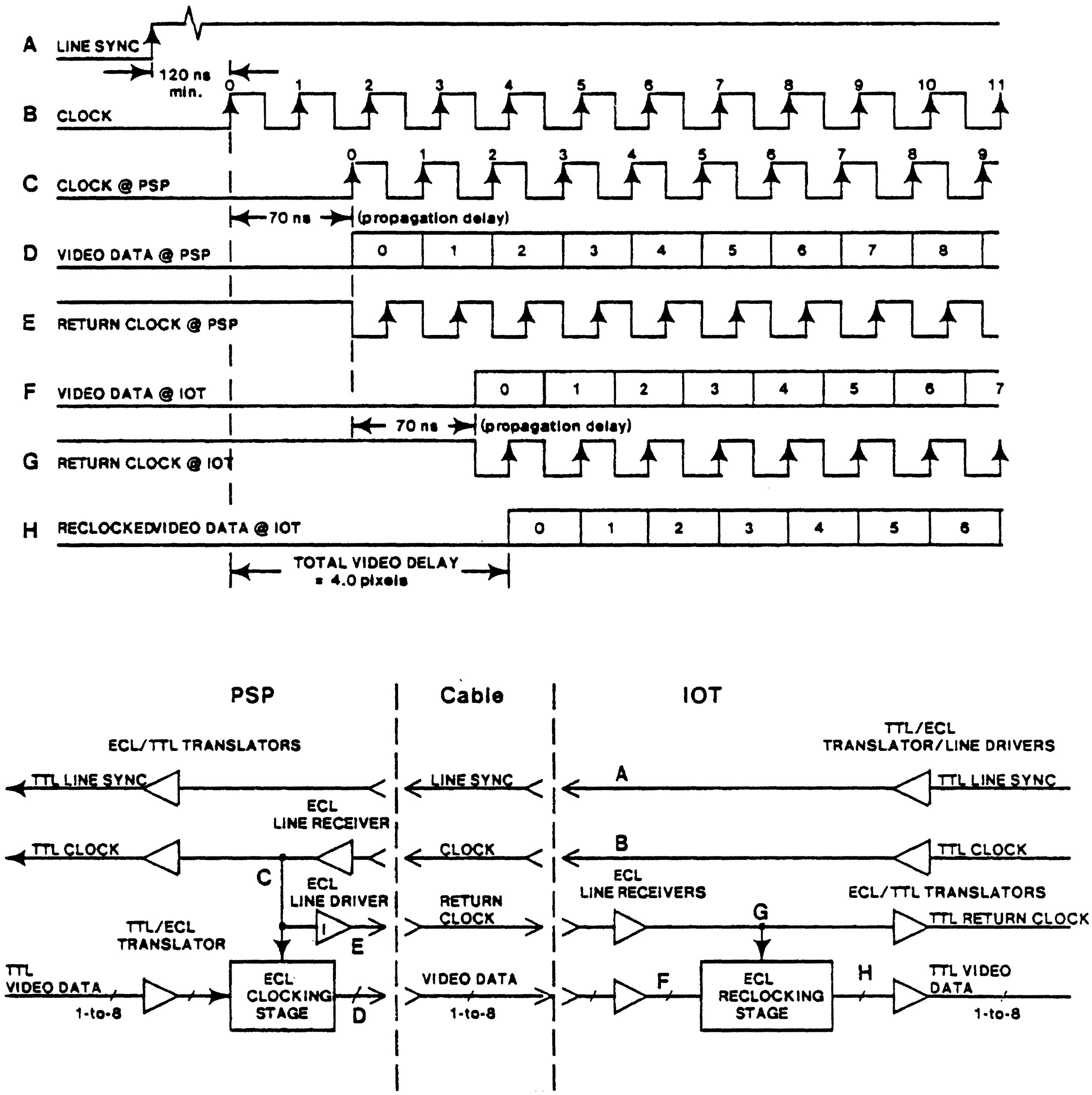
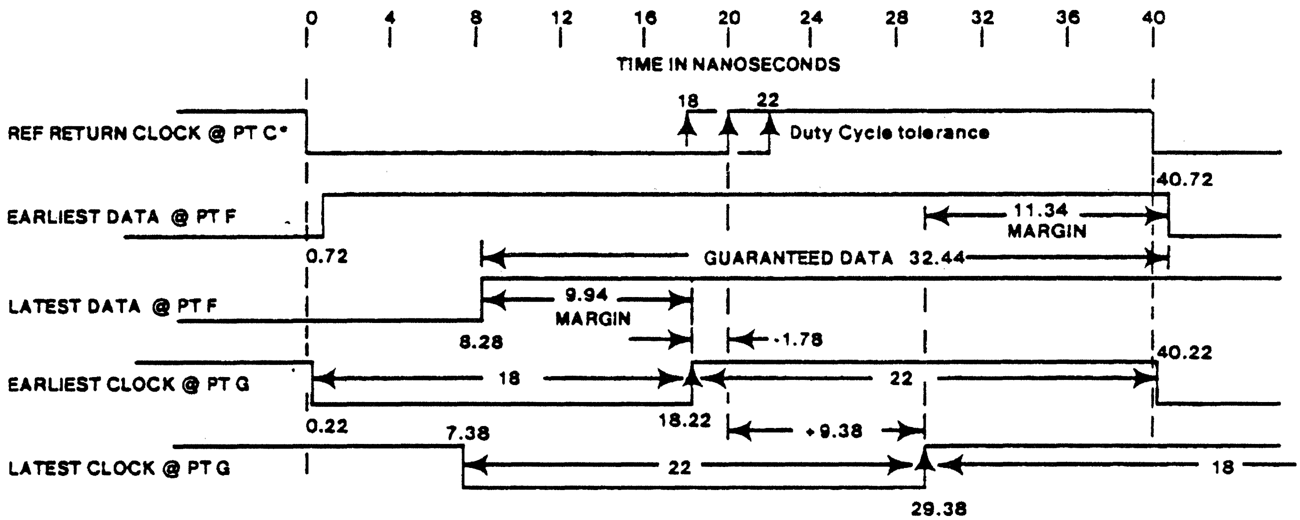


Figure 7-3. **Timing tolerance analysis for reclocked video data modes**



* See fig. 7.2
this is the ideal return clock referred to pt C, i.e. without the delay of the inverting line driver.

PARAMETER	VALUE	EARLIEST DATA	LATEST DATA	EARLIEST CLOCK	LATEST CLOCK
DATA RATE	25 MBPS (40 ns)				
CABLE LENGTH	50 ft				
VP	78%				
LINE DRIVER tp RANGE	1.0 to 2.8 ns			+1.0	+2.8
LINE RCVR tp RANGE	1.0 to 2.8 ns	+1.0	+2.8	+1.0	+2.8
CABLE DELAY TOLERANCE	+/-0.78 ns (+/-1%, 2% total)	-0.78	+0.78	-0.78	+0.78
PSP CLOCKING STAGE tp RANGE	1.5 to 3.7 ns	+1.5	+3.7		
CLOCK DUTY CYCLE TOLERANCE	+/-2.0 ns (+/-5%)			-2.0	+2.0
DATA JITTER	+/-1.0 ns (+/-2.5%, 5% p-p)	-1.0	+1.0		
CLOCK JITTER	+1.0 ns (+2.5%)			-1.0	+1.0
	TOTAL	+0.72	+8.28	-1.78	+9.38

8. Electrical characteristics of interface circuits

8.1 Line drivers

All line drivers shall be MECL 10K Series with both normal and inverted outputs for differential driving. The recommended components are gate MC10212 (Xerox Part No. 733W01732), line receiver MC10216 (XPN 733W02917) used as a line driver, flip-flop MC10231 (XPN 733W01734), and TTL/ECL translator MC10124 (XPN 733W01682). The generic circuit for line drivers is shown in figure 8-1. Other MECL 10K Series components with differential output may be used if different logic functionality is desired at the input, or if different functional modularity is desired. A timing analysis should be performed, however, if slower devices are substituted in the video or return clock circuits of the recommended interface implementation (refer to section 8.3).

8.2 Line receivers

All line receivers shall be MECL 10K Series with differential inputs. The recommended components are MC10216 (XPN 733W02917) and ECL/TTL translator MC10125 (XPN 733W01685). The generic circuit for line receivers is shown in figure 8-2. Other MECL 10K Series components with differential input may be used if different logic functionality is desired at the output, or if different functional modularity is desired. A timing analysis should be performed, however, if slower devices are substituted in the video or return clock circuits of the recommended interface implementation (refer to section 8.3).

8.3 Recommended circuit implementation

Circuits recommended for implementation of the PSP Imaging Interface and Command/Status Interface are shown in figures 8-3-A and 8-3-B. These circuits provide return byte (or pixel) clock, and accommodate both parallel and serial video data transmission modes. If only the serial mode is to be implemented, IOTclkECL-b is not needed, and only one flip-flop stage need be implemented. These circuits are the basis for the PSP side of the timing analysis model in section 7.2.

Circuits recommended for implementation of the IOT Imaging Interface and Command/Status Interface are shown in figures 8-4-A and 8-4-B. These circuits provide for use of return byte (or pixel) clock, and accommodate both parallel and serial video data transmission modes. For the serial video data mode, only one video data channel need be implemented, and RetClkECL-b is not needed. If the serial data is not to be reclocked in the IOT, the return pixel clock channel (below the dashed line in figure 8-4-A) need not be implemented. These circuits are the basis for the IOT side of the timing analysis model in section 7.2.

Note that the output of balanced line receivers may or may not be predictable under conditions of open input (cable disconnected) or signal source power off (consult device specifications). It is recommended that the output of line receivers be inhibited or ignored at a higher system level when such conditions exist. Circuits for detecting these conditions are discussed in section 8.5.

8.4 Power control

The required power control arrangement is shown in figure 8-5. A +12VDC power source in the PSP is connected to a low-power control actuator in the IOT via the **power control +** line in the interface cable. The control actuator, in turn, operates a power actuator which switches the AC mains in the IOT. The return circuit is via the **power control -** line in the interface cable and the control circuitry in the PSP. Resistor R4 limits the current which can be drawn from the +12VDC power source. The control actuator is not specified, but is nominally a 6V, 120 ohm device. Resistor R5 is selectable to allow a choice for the control actuator. It is considered to be part of the load in the IOT. The design of this power control arrangement, the recommended control circuitry for the PSP, and an example selection of R5 are discussed in appendix D. The power source must be capable of supplying at least 65 ma operationally. Resistance R4 must have an effective power rating sufficient to withstand a short circuit to ground of the **power control +** line, without damage to itself or to the printed wire board. The power source must be able to supply worst case short circuit current of 150 ma (5 percent high voltage, 9 percent low R4).

The PSP Power Normal input to the control circuitry must detect AC power line failures greater than 10 milliseconds, and must drop **power control** to the off condition within 50 milliseconds and hold it off for at least 2 milliseconds. The control actuator device in the IOT, and the control circuitry in the PSP, must be protected from transients as large as 20V and as long as 1 microsecond on the **power control** lines (a filtering scheme for this purpose is shown in appendix D). The control actuator device in the IOT must also be safety isolated from the AC power source to at least 1600 VAC. It is expected that the Vcc power supplies in the PSP and the IOT will continue to supply regulated output for 5 milliseconds after AC power has been removed from their input by a power failure or by switching.

The control circuitry in the PSP must be glitch-free during power turn-on and turn-off. That is, until the Power Normal signal *and* the Power On signal are both at the high-level state, (> 2.4 v), the control output device shall be an open circuit. It is also recommended that the origin of the Power On signal be protected by use of encoded turn-on and turn-off signals, rather than simple binary state changes.

Whenever a non-emergency IOT power-down is initiated via the control element in the PSP, the PSP must send an advanced warning to the IOT via Client Layer message, *PspMetaReset*. The amount of time required for the advanced warning is an IOT parameter, which must be conveyed to the associated PSP as the *PowerDownWarning* parameter of the Client Layer message, *IotConfiguration*. If the time is less than 5 milliseconds, no warning message need be sent. The default time is zero.

Results of activating or deactivating the Power On signal may be monitored in the PSP by means of the detection circuit discussed in section 8.5.

8.5 Detection of cable and power status at the interface

Certain interface conditions can be detected at the line receivers, namely, power on/off at the opposite side of the interface, and/or the interface cable disconnected. Figure 8.6 shows a simple addition to the line receiver circuit to accomplish this.

In figure 8-6, the upper part of the diagram is the normal cable termination and line receiver circuit (see, for example, figure 8-3-A). When the signal source is powered and connected, the voltage at point A in the diagram will be at the midpoint of the ECL signal range, i.e., about −1.29 vdc. This will be true whether or not the signal is active, and for either quiescent logic state. Each MC125 detector is biased by the voltage divider, relative to point A, so that its output is logic low. If the cable is disconnected, point A goes to approximately −5.2 vdc. This biases the upper detector further off, but biases the lower detector so that its output is at the high logic level. If the cable is connected but the power on the other side of the interface is off, the signal source becomes a low impedance to ground, and point A goes to near ground. This biases the lower detector further off, but biases the upper detector so that its output is at the high logic level.

These circuits are optional in the PSP for monitoring the normal IOT power on/off state, and also to provide low-level interface diagnostics. It is recommended that, if used, they be implemented on the **status** line receiver. They may also be used in the IOT for diagnostic purposes. In the IOT, they should be implemented on the command line receiver. The cable-disconnected detector may also be used to accomplish hardware or software override of any line receiver output which causes a problem when the PSP or IOT must retain some level of functionality when disconnected at this interface.

Either or both of the detectors can be implemented, as desired, and they may be implemented on one or more of the line receivers. Theoretically, they do not affect the performance of the signal path. It is recommended, however, that they not be applied to a high-speed signal path without verifying the performance.

8.6 Interface power supplies

The dc power sources serving the interface ECL circuits in the PSP and in the IOT shall be −5.2 vdc ±5 percent. The power supply common in each unit shall be signal ground for the interface circuits and shall be connected to the common logic ground within the respective units.

8.7 Cables

When the generic interface is implemented for serial video data transmission, the 9-pair cable specified by Xerox Drawing No. 117P23905 shall be used. When the generic interface is implemented for parallel video data transmission, the 16-pair cable specified by Xerox Drawing No. 117P23904 shall be used. These drawings define cables composed of individually-shielded twisted pairs, each with nominal characteristic impedance (Zo) of 100 ohms, and nominal velocity of propagation (Vp) of 0.78c. All of the shielded pairs are enclosed within a single overall shield. Each signal of the generic interface shall be carried on one pair of the cable. Each cable provides one spare shielded twisted pair. The complete electrical and mechanical specifications are contained in the drawings.

8.8 Connectors

When the generic interface is implemented for serial video data transmission, connectors, receptacles, and auxilliary hardware from the 37-pin D-Shell series shall be used. When the generic interface is implemented for parallel video data transmission, connectors, receptacles, and auxiliary hardware from the 50-pin D-Shell series shall be used (part numbers are given in chapter 9).

8.9 Shielding and grounding (refer to figure 8-7)

Within the PSP and within the IOT, the cable shields shall be connected as follows:

a) The outer shield of the interface cable shall be connected to a ground "mecca" that is a common point for frame ground and green wire ground of the AC power source. Connection is to be via both pin 1 and the connector shell. It is desirable that the outer shield ground circuit be completed before any other circuit is completed, when the cable is installed.

b) The shield circuit of each twisted pair shall be connected to signal ground (logic ground).

c) Provision must be made for connecting signal ground to the ground mecca within the PSP and within the IOT; however, the nature (impedance) of the connection, including no connection, is to be determined by the product program.

d) The common side of the −5.2v power supply serving the interface drivers and receivers must be tied to signal ground within each unit.

The length of the balanced signal-runs inboard of the interface connector should be minimized. Etch runs should be in the form of differential microstrip or differential stripline, with characteristic impedance of 100 ohms to match the characteristic impedance of the shielded twisted pairs. The specific physical arrangements for cable routing and grounding within the PSP and IOT are the responsibility of the respective product programs.

Figure 8-1. **Generic circuit for line drivers**

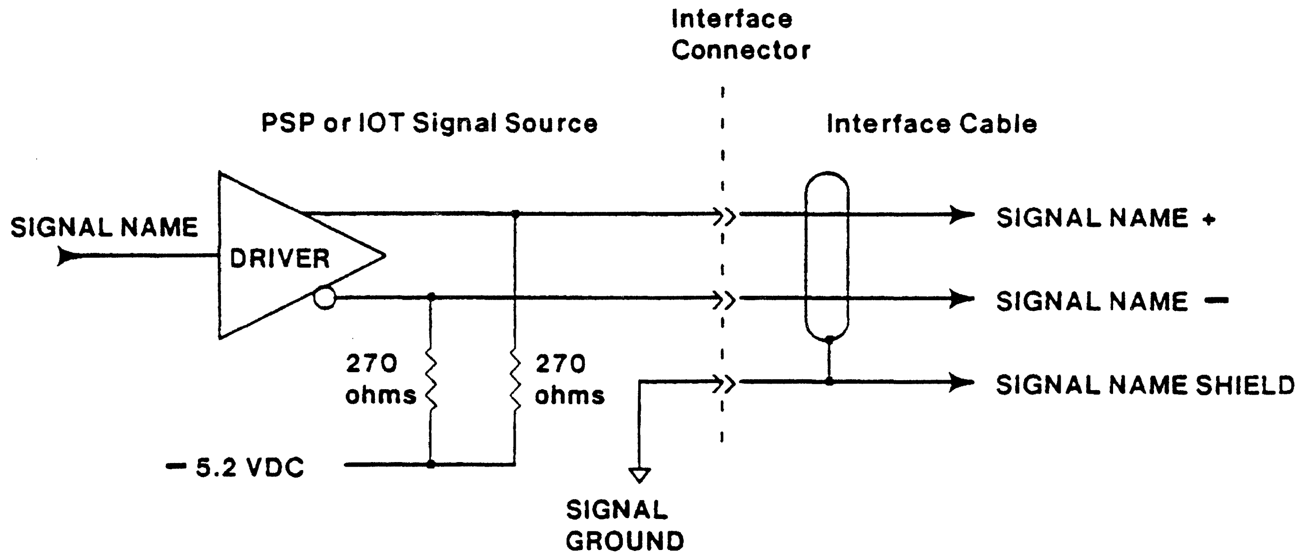


Figure 8-2. **Generic circuit for line receivers**

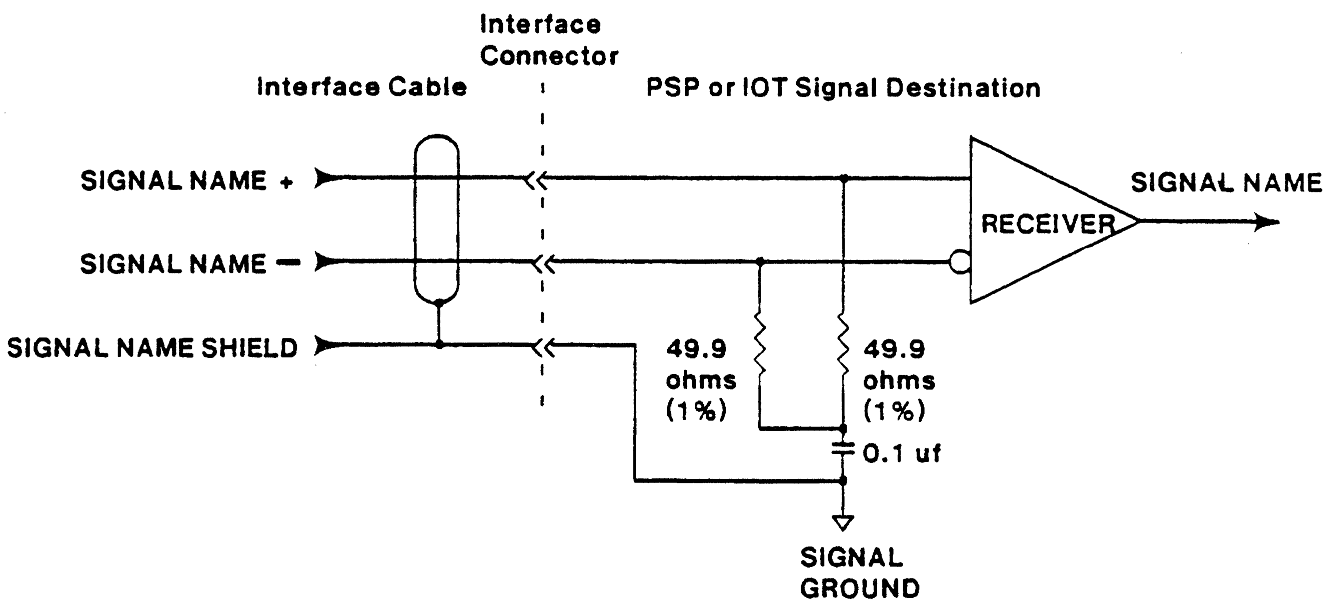


Figure 8-3-A. **Circuits recommended for implementation of PSP imaging interface and Command/Status interface.** Both parallel and serial video data modes are provided.

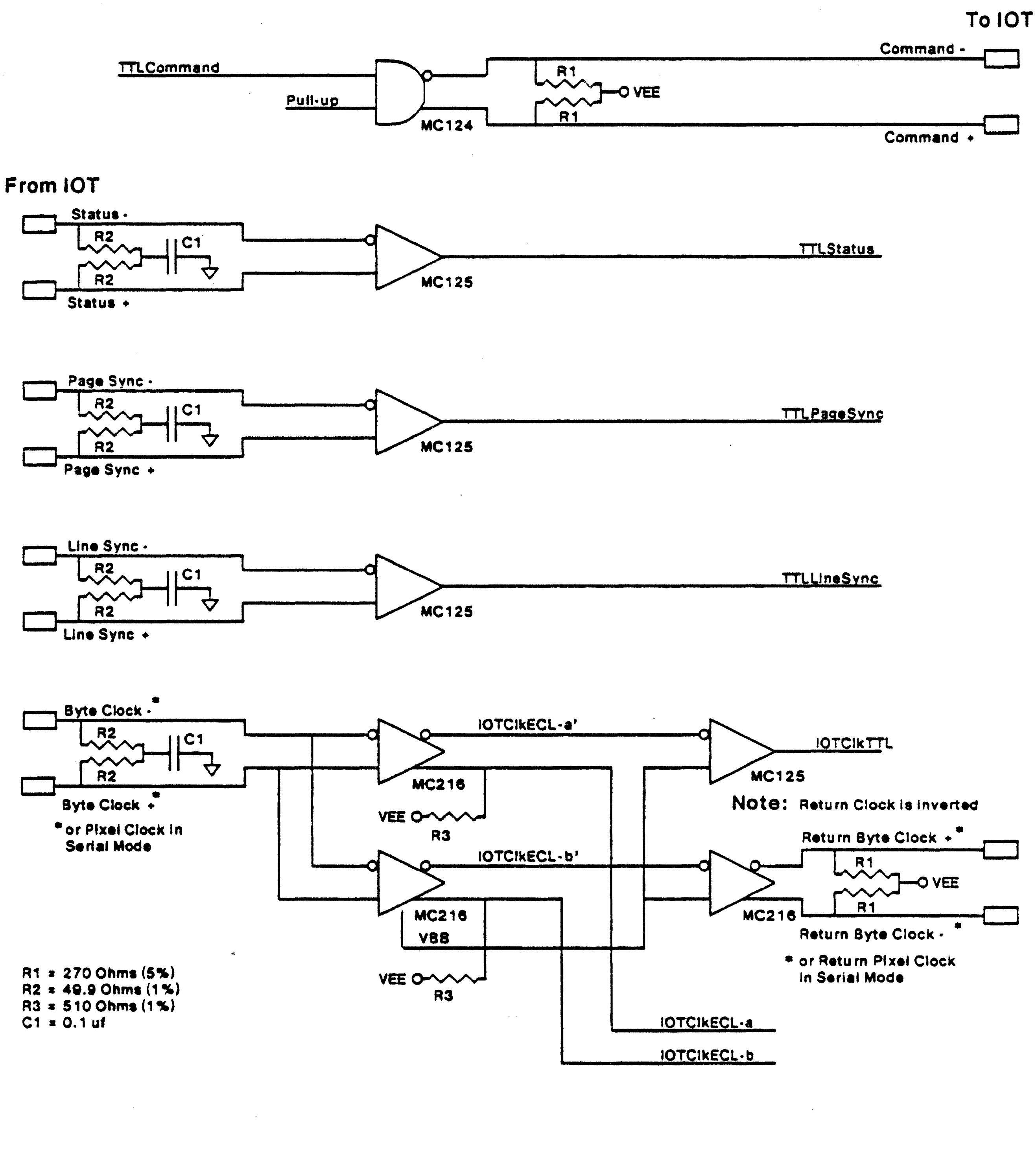


Figure 8-3-B. **Circuits recommended for implementation of PSP video data interface.** Both parallel and serial video data modes are provided.

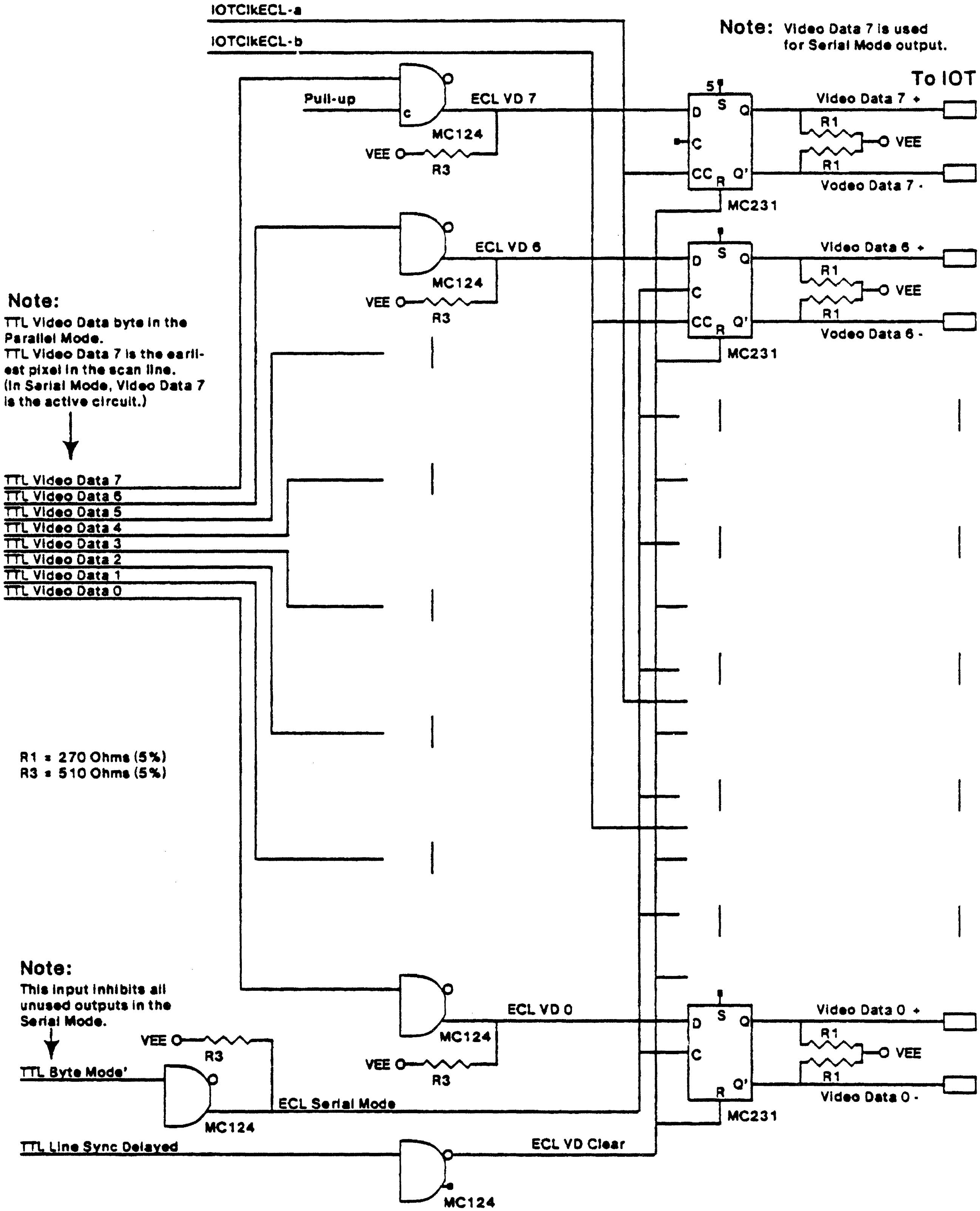


Figure 8-4-A. **Circuits recommended for implementation of IOT imaging interface and Command/Status interface.** Both parallel and serial video data modes are provided.

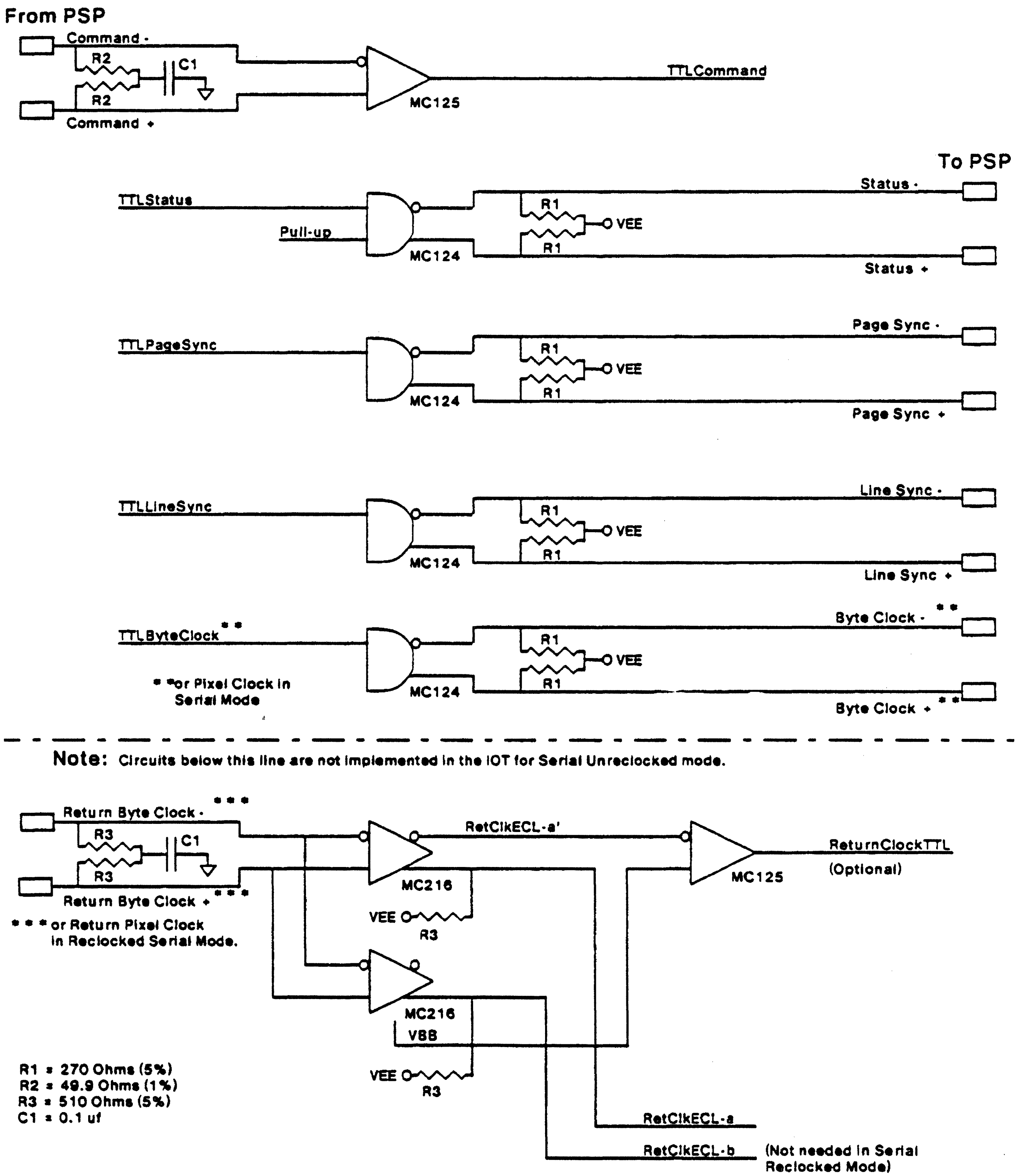


Figure 8-4-B. **Circuits recommended for implementation of IOT video data interface, parallel mode.** (Only circuits associated with Video Data 7 are implemented for Serial Reclocked Mode.)

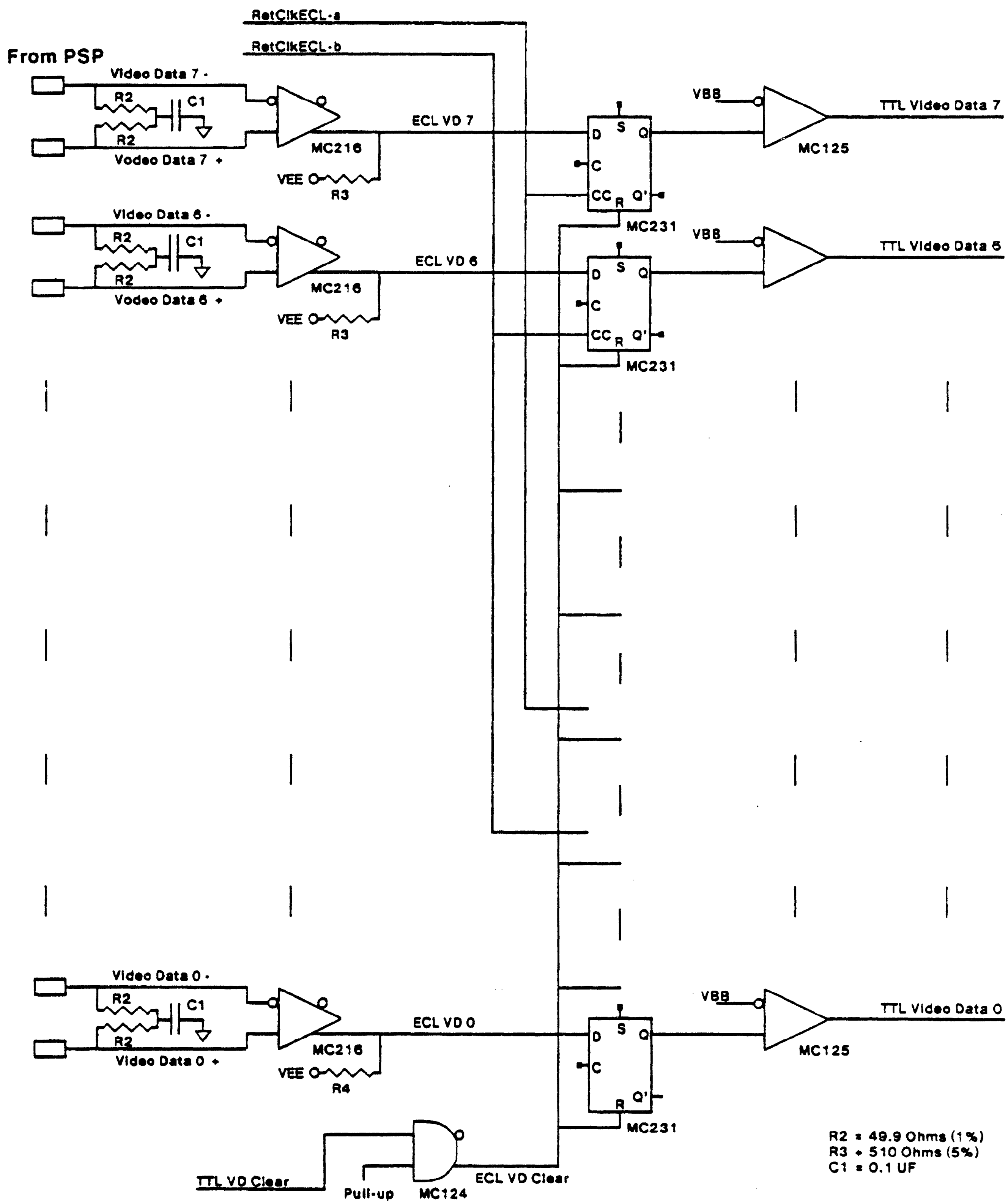
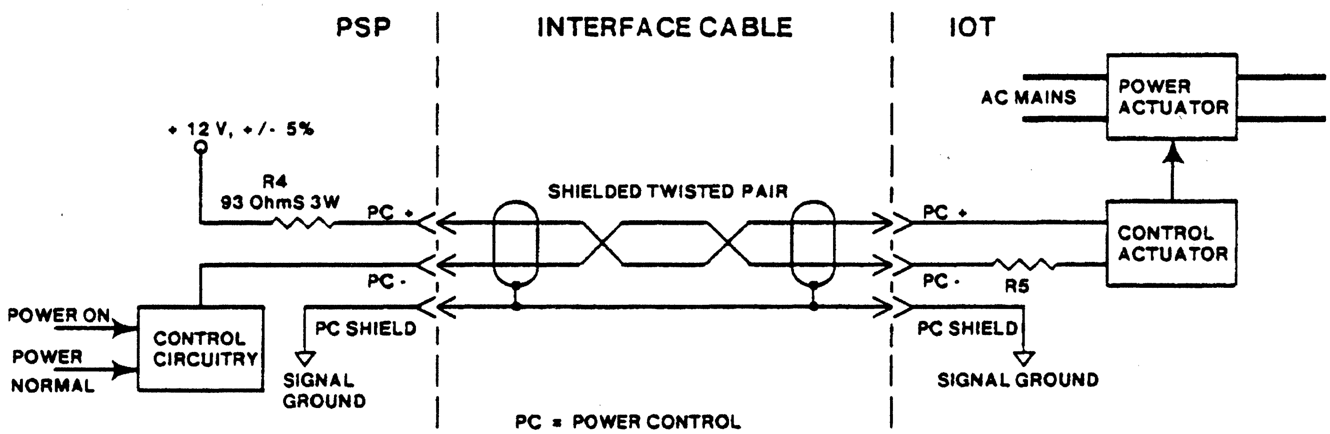
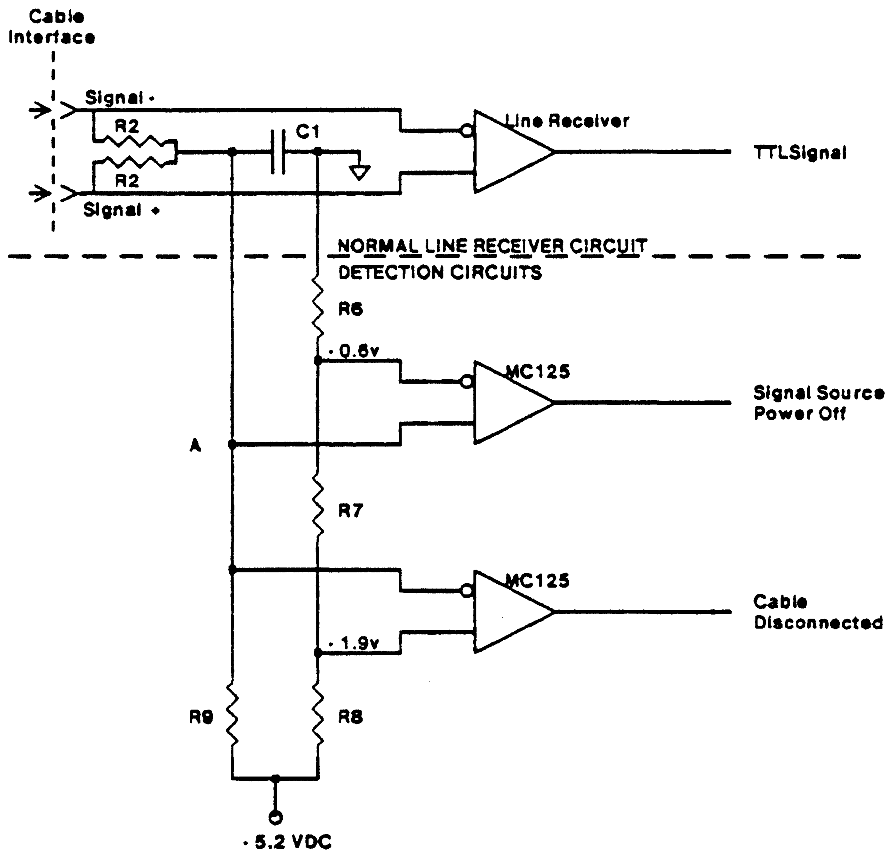


Figure 8-5. **Power control interface**
(also refer to figure D-1, appendix D.)



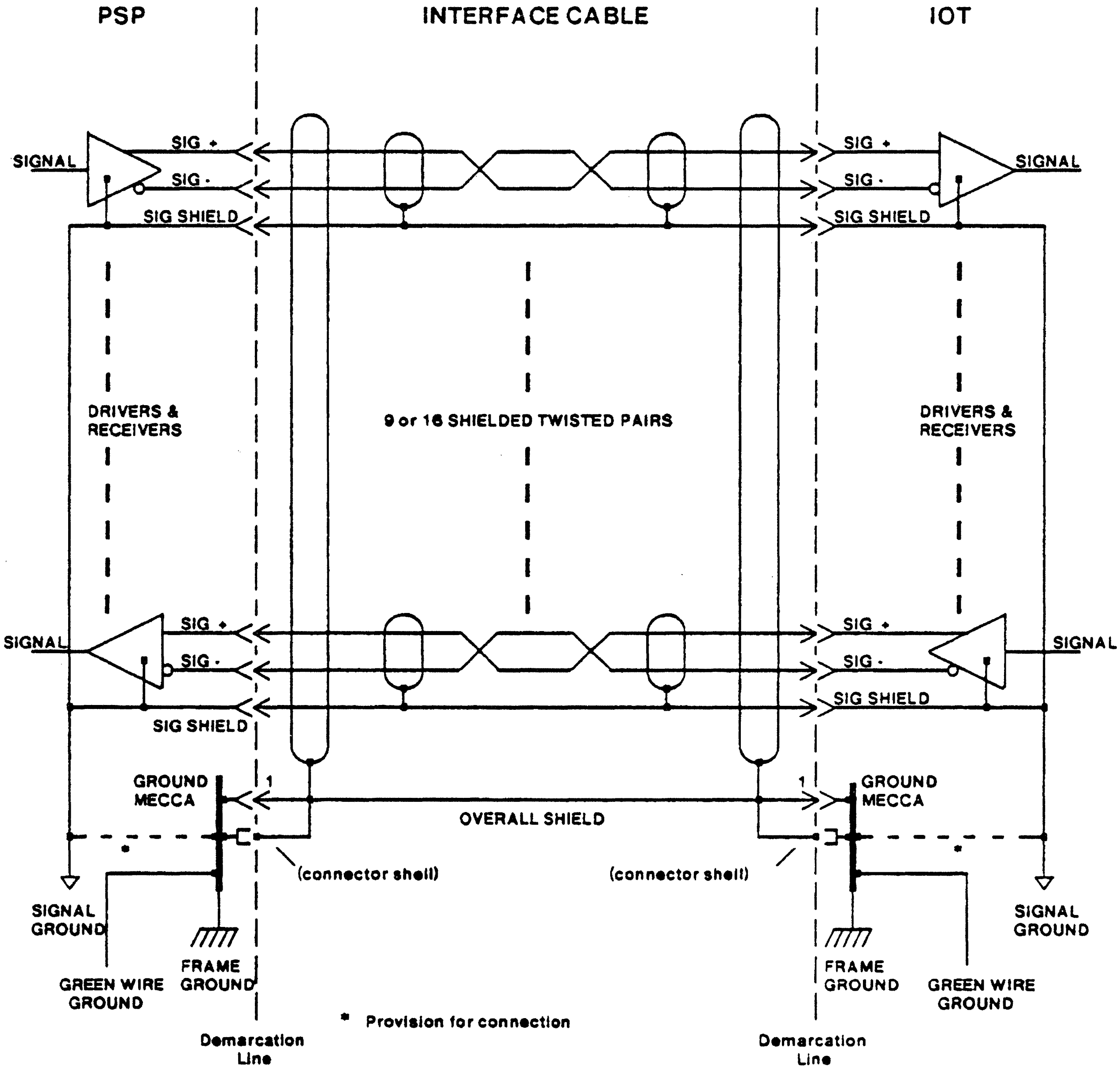
R2 is IOT specific, based on Control Actuator used.

Figure 8-6. **Circuits for detection of cable and power status at the interface**



R2 = 49.9 Ohms
R6 = 620 Ohms
R7 = 1.3K Ohms
R8 = 3.3K Ohms
R9 = 47K Ohms
C1 = 0.1 uf

Figure 8-7. **Generic grounding arrangements**



9. Mechanical characteristics of interface

The hardware recommended below is the most appropriate currently available. Various product programs are working to develop compatible improvements (for example, less bulky, molded, better shielding). As these become available, they will be added to this generic specification.

9.1 Demarcation points (refer to figures 8-6 and 9-2)

The PSP interface with the IOT is defined to be at the mating point of the 37-pin (or 50-pin) receptacle mounted in the PSP, with the matching plug mounted on the interface cable. The receptacle in the PSP must be mounted on a backpanel (or equivalent), with no more than one PWB edge connector between this receptacle and the active components of the Imaging Interface circuits.

The IOT interface with the PSP is defined to be at the mating point of the 37–pin (or 50–pin) receptacle mounted in the IOT, with the matching plug mounted on the interface cable. The receptacle in the IOT must be mounted on a backpanel (or equivalent) with no more than one PWB edge connector between this receptacle and the active components of the Imaging Interface circuits. The shield of this receptacle must be connected to the IOT ground "mecca" via its mounting hardware.

9.2 Connectors

9.2.1 37-pin D-Shell

The following connectors and accessory parts from the 37-pin D-Shell series shall be used, as appropriate, in the PSP, the IOT, and the interconnecting cable assembly, when the generic interface is implemented for serial video data transmission with the 9-pair cable described in section 8.4.

Description	Xerox Drawing No.
Plug (for cable assembly)	713W02637
Contact (male, for plug)	713W00937
Shell (for plug)	118P20514
Outer Ferrule (for shell-cable assembly)	115P00638
Inner Ferrule (for shell-cable assembly)	115P00669
Receptacle (vertical PWB mount, pre-pinned, female, pressfit)	713W20237
Screw-lock (for PWB mount receptacle)	713W81237
Receptacle (bulkhead-mount)	713W02737
Contact (female, for bulkhead-mount receptacle)	713W01037
Screw-lock (for bulkhead-mount receptacle)	713W80737

9.2.2 50-pin D-Shell

The following connectors and accessory parts from the 50-pin D-Shell series shall be used, as appropriate, in the PSP, the IOT, and the interconnecting cable assembly when the generic interface is implemented for parallel video data transmission with the 16-pair cable described in section 8.7. This connector will also be used in other special applications (refer to sections 9.2.4 and 9.3).

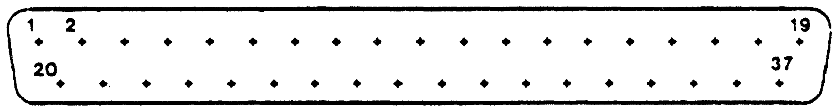
Description	Xerox Drawing No.
Plug (for cable assembly)	713W02837
Contact (male, for plug)	713W00937
Shell (for plug)	118P20520
Outer Ferrule (for shell-cable assembly)	115P00639
Inner Ferrule (for shell-cable assembly)	115P00672
Inner Ferrule (for shell-cable assembly with 9-pair cable)	115P00673
Receptacle (vertical PWB mount, pre-pinned, female, pressfit)	713W20337
Screw-lock (for PWB mount receptacle)	713W81237
Receptacle (bulkhead-mount)	713W02937

Description	Xerox Drawing No.
Contact (female, for bulkhead-mount receptacle)	713W01037
Screw-lock (for bulkhead-mount receptacle)	713W80737

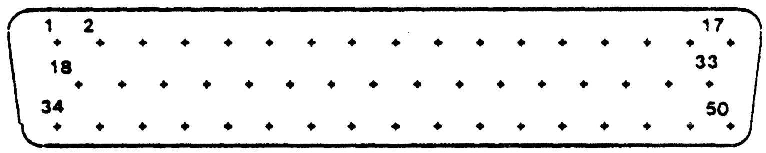
9.2.3 Pin orientation

The physical layout of the pins in the 37-pin and 50-pin connector series is shown in figure 9-1.

Figure 9-1. **Connector pin orientations** (viewing face of plug)



37-PIN SERIES



50-PIN SERIES

9.2.4 Pin assignments

a) **37-pin, Serial Video Mode:**

Pin No.	Signal	Pin No.	Signal
1	Frame Ground		
2	Power Control +	20	Power Control –
3	Signal Ground	21	Signal Ground
4	Spare +	22	Spare –
5	Signal Ground	23	Signal Ground
6	Command +	24	Command –
7	Signal Ground	25	Signal Ground
8	Status +	26	Status –
9	Signal Ground	27	Signal Ground
10	Page Sync +	28	Page Sync –
11	Signal Ground	29	Signal Ground
12	Line Sync +	30	Line Sync –
13	Signal Ground	31	Signal Ground
14	Pixel Clock +	32	Pixel Clock –
15	Signal Ground	33	Signal Ground

Pin No.	Signal	Pin No.	Signal
16	Return Pixel Clock +	34	Return Pixel Clock –
17	Signal Ground	35	Signal Ground
18	Video Data +	36	Video Data –
19	Signal Ground	37	Signal Ground

b) **50-pin, Parallel Video Mode:**
(Also serves Serial Mode in universal PSPs with Video Data 0 through 6 not used.)

Pin No.	Signal	Pin No.	Signal	Pin No.	Signal
1	Frame Ground				
2	Power Control +	18	Signal Ground	34	Spare +
3	Power Control –	19	Signal Ground	35	Spare –
4	Command +	20	Signal Ground	36	Status +
5	Command –	21	Signal Ground	37	Status –
6	Byte Clock +	22	Signal Ground	38	Line Sync +
7	Byte Clock –	23	Signal Ground	39	Line Sync –
8	Page Sync +	24	Signal Ground	40	Return Byte Clock +
9	Page Sync –	25	Signal Ground	41	Return Byte Clock –
10	Video Data 1 +	26	Signal Ground	42	Video Data 0 +
11	Video Data 1 –	27	Signal Ground	43	Video Data 0 –
12	Video Data 3 +	28	Signal Ground	44	Video Data 2 +
13	Video Data 3 –	29	Signal Ground	45	Video Data 2 –
14	Video Data 5 +	30	Signal Ground	46	Video Data 4 +
15	Video Data 5 –	31	Signal Ground	47	Video Data 4 –
16	Video Data 7 +	32	Signal Ground	48	Video Data 6 +
17	Video Data 7 –	33	Signal Ground	49	Video Data 6 –
				50	

9.3 Cable assemblies

The product programs must design and issue cable assemblies according to their requirements, using the plugs, shells, and auxiliary parts prescribed in section 9.2, together with 9-pair cable, Xerox Part No. 117P23905, or 16-pair cable, Xerox Part No. 117P23904 (also refer to section 8.7), and using the following guidelines.

9.3.1 Cable assembly types

There are three basic cable assemblies:

Type-A 37-pin plug on each end of 9-pair cable (both PSP and IOT designed for serial video data).

Type-B 50-pin plug on each end of 16-pair cable (both PSP and IOT designed for parallel video data).

Type-C 50-pin plug on PSP end, 37-pin plug on IOT end of 9-pair cable (PSP designed for parallel video data, and IOT designed for serial video data).

9.3.2 Pin assignments

a) **Type-A cable assemblies shall use the following pin assignments for each 37-pin plug:**

Pin No.	Signal	Pin No.	Signal
1	Overall shield[1]		
2	Power Control +	20	Power Control –
3	Power Control Shield	21	
4	Spare +	22	Spare –
5		23	Spare Shield
6	Command +	24	Command –
7	Command Shield	25	
8	Status +	26	Status –
9		27	Status Shield
10	Page Sync +	28	Page Sync –
11	Page Sync Shield	29	
12	Line Sync +	30	Line Sync –
13		31	Line Sync Shield
14	Pixel Clock +	32	Pixel Clock –
15	Pixel Clock Shield	33	
16	Return Pixel Clock +	34	Return Pixel Clock –
17		35	Return Pixel Shield
18	Video Data +	36	Video Data –
19	Video Data Shield	37	

[1] Shield must be terminated to connector shell as well as to pin 1, and must not be connected to any other pin.

b) **Type-B cable assemblies shall use the following pin assignments for each 50-pin plug:**

Pin No.	Signal	Pin No.	Signal	Pin No.	Signal
1	Overall shield[1]				
2	Power Control +	18	Power Control Shield	34	Spare +
3	Power Control −	19	Spare Shield	35	Spare −
4	Command +	20	Command Shield	36	Status +
5	Command −	21	Status Shield	37	Status −
6	Byte Clock +	22	Byte Clock Shield	38	Line Sync +
7	Byte Clock −	23	Line Sync Shield	39	Line Sync −
8	Page Sync +	24	Page Sync Shield	40	Return Byte Clock +
9	Page Sync −	25	Return Byte Clock Shield	41	Return Byte Clock −
10	Video Data 1 +	26	Video Data 0 Shield	42	Video Data 0 +
11	Video Data 1 −	27	Video Data 1 Shield	43	Video Data 0 −
12	Video Data 3 +	28	Video Data 2 Shield	44	Video Data 2 +
13	Video Data 3 −	29	Video Data 3 Shield	45	Video Data 2 −
14	Video Data 5 +	30	Video Data 4 Shield	46	Video Data 4 +
15	Video Data 5 −	31	Video Data 5 Shield	47	Video Data 4 −
16	Video Data 7 +	32	Video Data 6 Shield	48	Video Data 6 +
17	Video Data 7 −	33	Video Data 7 Shield	49	Video Data 6 −
			50		

c) **Type-C cable assemblies shall use the Type-A pin assignments given above for the 37-pin plug, and shall use the pin assignments given below for the 50-pin plug.**

Pin No.	Signal	Pin No.	Signal	Pin No.	Signal
1	Overall shield[1]				
2	Power Control +	18	Power Control Shield	34	Spare +
3	Power Control −	19	Spare Shield	35	Spare −
4	Command +	20	Command Shield	36	Status +
5	Command −	21	Status Shield	37	Status −
6	Pixel Clock +	22	Pixel Clock Shield	38	Line Sync +
7	Pixel Clock −	23	Line Sync Shield	39	Line Sync −
8	Page Sync +	24	Page Sync Shield	40	Return Pixel Clock +
9	Page Sync −	25	Return Pixel Clock Shield	41	Return Pixel Clock −
10		26		42	
11		27		43	

[1] Shield must be terminated to connector shell as well as to pin 1, and must not be connected to any other pin.

Pin No.	Signal	Pin No.	Signal	Pin No.	Signal
12		28		44	
13		29		45	
14		30		46	
15		31		47	
16	Video Data +	32		48	
17	Video Data -	33	Video Data Shield	49	
				50	

9.3.3 Cable length, routing and handling

Cable assemblies may be any length up to 50 feet maximum. The length of the cable is measured from the PSP interface to the IOT interface, as the interfaces are defined in section 9.1 above, i.e., the plugs on each end are included.

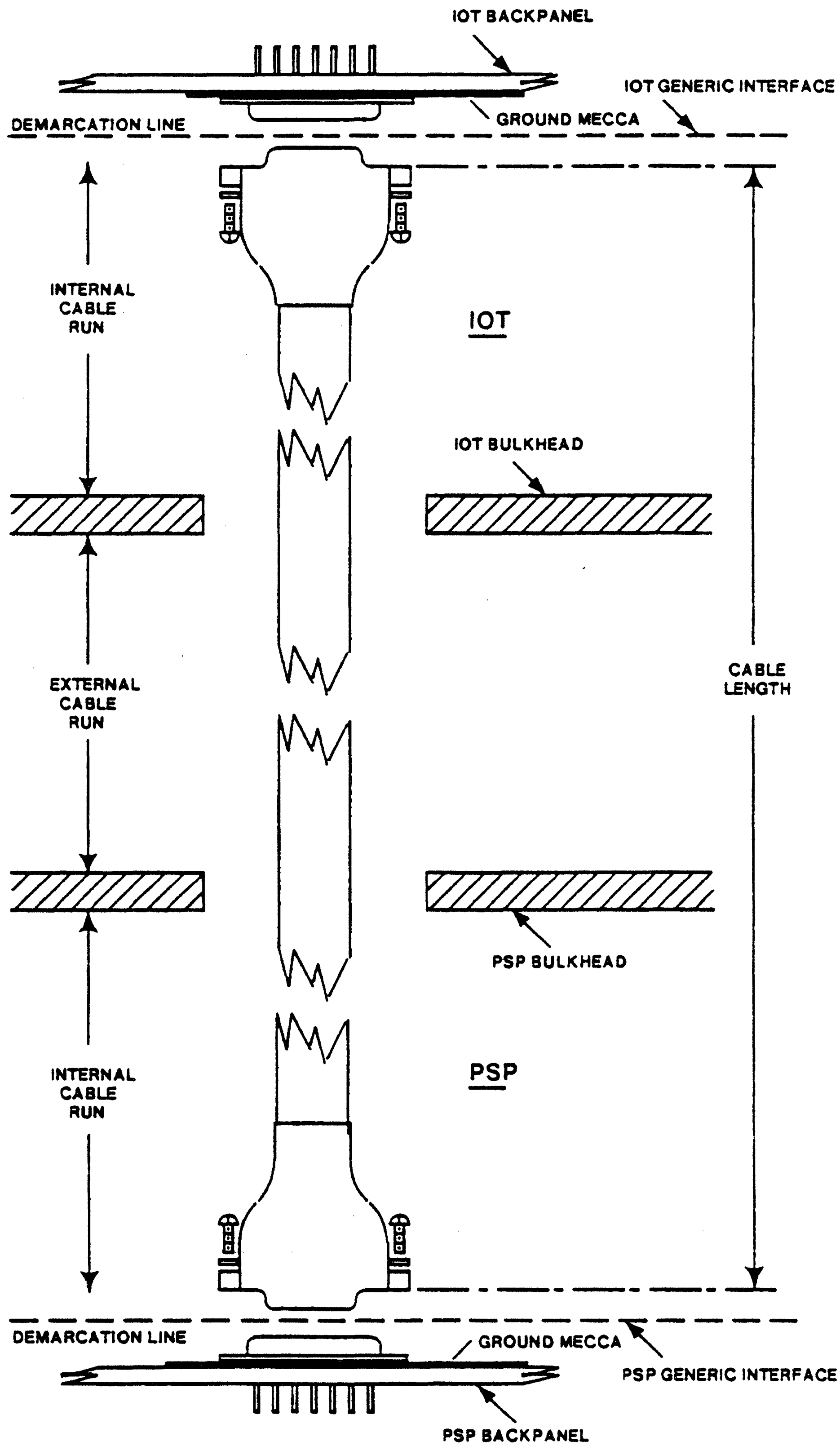
Internal or external routing and securing must not subject either cable to a bend radius of less than five times its maximum diameter.

The maximum diameter of the 9-pair cable is 0.595 inches, and the maximum diameter of the 16-pair cable is 0.650 inches. A bulkhead hole approximately 0.69 x 2.88 inches is required to pass through the 37-pin plug. A bulkhead hole approximately 1.0 x 2.75 inches is required to pass through the 50-pin plug.

The overall shield is covered by a protective insulated jacket, and must be grounded only through the shells and the assigned pins of the plugs and their mating receptacles. It must not be grounded at any other point in the cable run.

The prescribed cables will have the UL2919 rating. They are *not* plenum rated.

Figure 9-2. **Physical location of generic interfaces**



10. Standards agency compliance

10.1 Safety

The presecribed interface has been designed with the express intent that incorporation into a product will not jeopardize compliance of that product with the following safety reguirements:

UL 478
CSA C22.2 No. 154
IEC 950 (This new standard supersedes IEC 380 and IEC 435.)

The cables prescribed herein will have UL2919 rating. They are not plenum rated.

The interface prescribed herein will not provide power isolation per VDE 0804/572 and BPO Technical Guide No. 26, which is required by certain European PTTs. Such isolation must be provided in the PSP power supplies and in the IOT power supplies, or by means of special isolation arrangements at the telephone line interface in the PSP, or by a specially-designed PSP/IOT interface.

Each product program remains responsible for compliance with the appropriate safety standards.

10.2 Electro-magnetic emissions (EME)

The prescribed interface has been designed with the express intent that incorporation into a product will not jeopardize compliance of that product with the following EME requirements; however, each product must individually demonstrate compliance:

FCC Subpart J Part 15 Level A
VDE 0871 Level A
VDE 9875
82/499/EEC[1]

([1] A new emission standard has been prepared by CISPR and published as CISPR 22. It is anticipated that within the next 18 months, it will replace 82/499/EEC and may also be adopted by the FCC. This new standard has two classes of limits, similar to current VDE and FCC specifications. 5/13/86.)

Each product program remains responsible for compliance with the appropriate emission standards.

A. References

[1] International Organization for Standardization. *ISO 3309, Data Communication–High-level Data Link Control Procedures—Frame Structure*. (Second Edition—1979-07-01).

[2] International Organization for Standardization. *ISO 4335, Data Communication–High-level Data Link Control Procedures—Elements of Procedures*. (First Edition—1979-04-15) with Addendum 1, 1979-12-15 and Addendum 2, 1982.

[3] International Organization for Standardization. *ISO 6159, Data Communication–HDLC Unbalanced Class of Procedures*. (First Edition-1980-05-01).

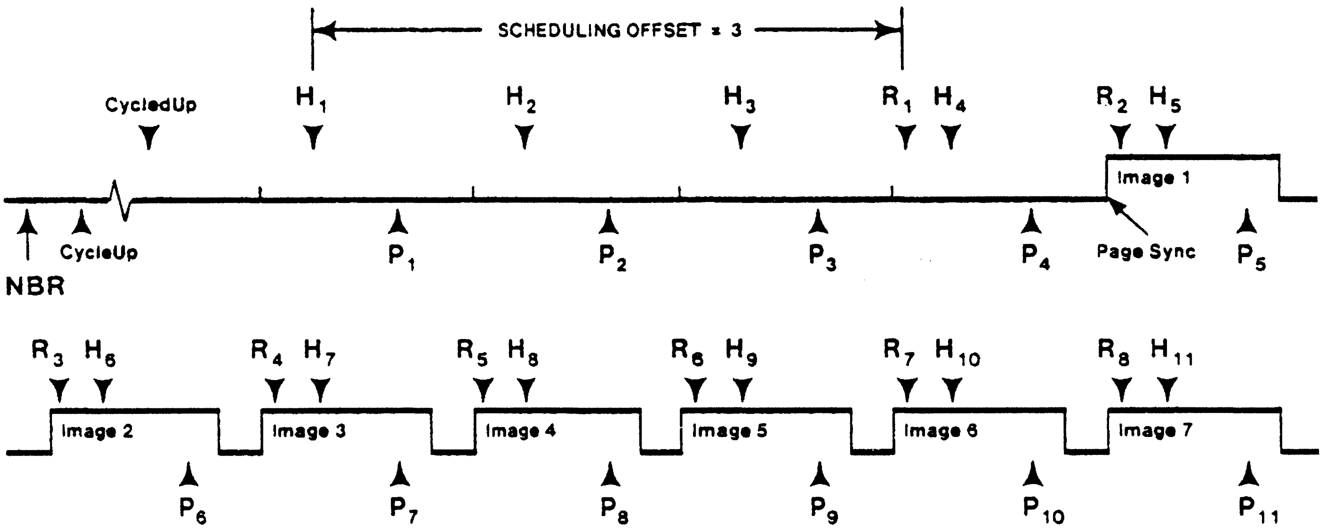
[4] IBM System Communications Division. *IBM Synchronous Data Link Control, General Information*. GA27-3093-2, File No. GENL-09. Research Triangle Park, North Carolina, 1979.

[5] International Organization for Standardization. *ISO 646, 7-bit Coded Character Set for Information Processing Interchange*. (First Edition—1973-07-1)

B. Operating sequence examples, scheduling offset of 3

Following are seven of the eight operating sequence examples given in section 6.5, with scheduling offset equal to 3 instead of 1.

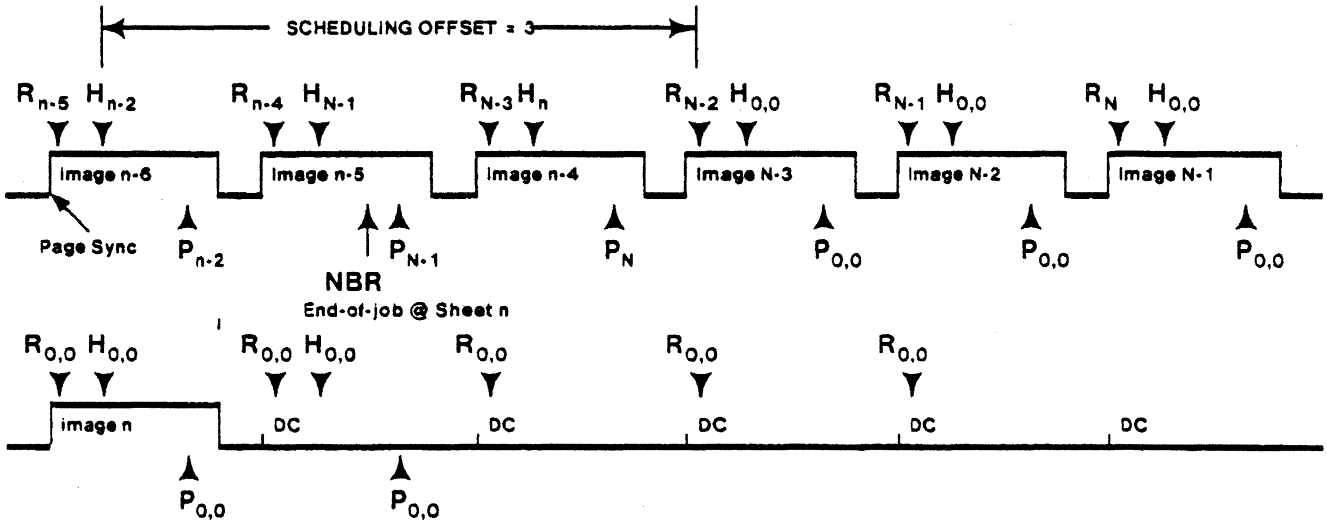
Figure B6-7-a. **Simplex cycle-up sequnce (scheduling offset = 3)**



$R_{0,0}$ ::= Request for a Dead Cycle (DC)

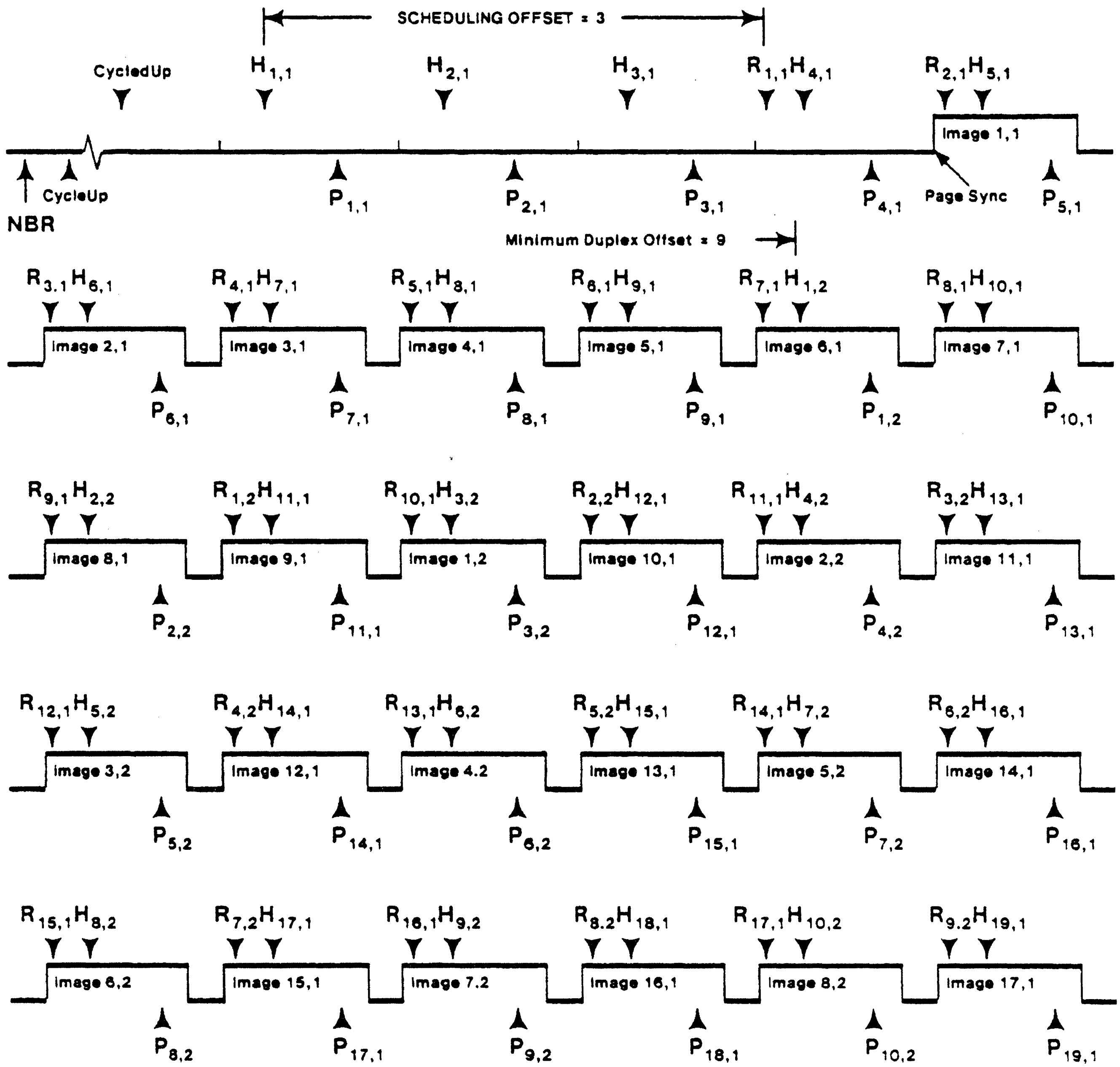
H_n P_n R_n ::= n-th "Hint, Print, Request" triplet. (Only sheet nos. are shown.)

Figure B6-7-b. **Simplex cycle–down sequnce**
(scheduling offset = 3)



$H_{0,0}$ ::= "Hint" Dead Cycle (DC)
$P_{0,0}$ ::= Print Dead Cycle
$R_{0,0}$ ::= Request for a Dead Cycle
H_n P_n R_n ::= n-th "Hint, Print, Request" triplet. (Only sheet nos. are shown.)
NBR ::= NextBankRequest

Figure B6-8. **Duplex cycle-up sequence**
(scheduling offset = 3)
(Storing duplex paper path, minimum duplex offset = 9)



$H_{i,j}$ $P_{i,j}$ $R_{i,j}$ ::= "Hint, Print, Request" triplet for sheet i, plate j.

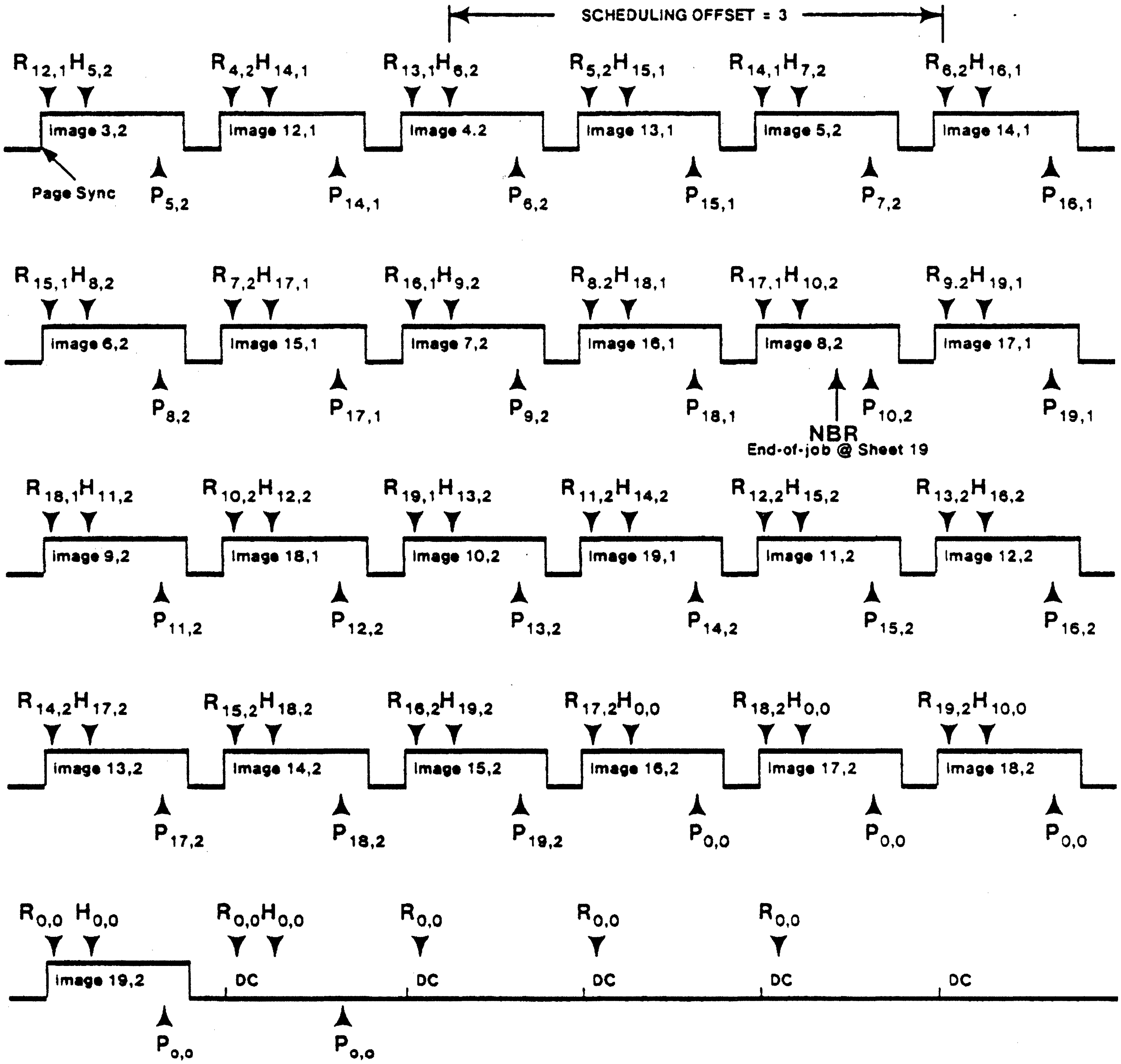
j ::= PlateNumber

PlateNumber ::= simplex side | duplex side

simplex side ::= 1

duplex side ::= 2

Figure B6-9. **Duplex cycle-down sequence**
(scheduling offset = 3)
(Storing duplex paper path, minimum duplex offset = 9)



$H_{i,j}\ P_{i,j}\ R_{i,j}$ ::= "Hint, Print, Request" triplet for sheet i, plate j.

j ::= PlateNumber

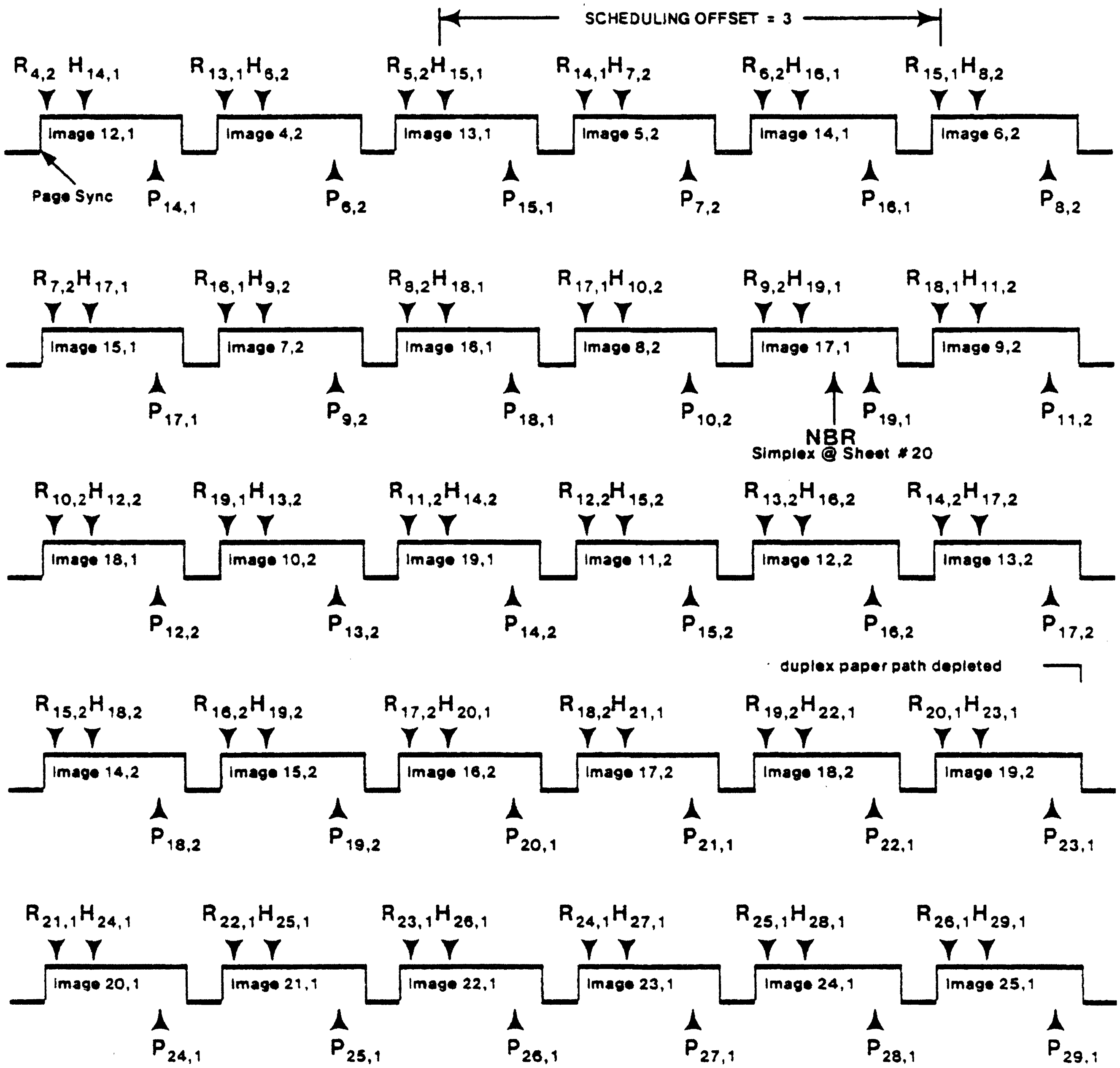
PlateNumber ::= simplex side | duplex side

simplex side ::= 1

duplex side ::= 2

NBR ::= NextBankRequest

Figure B6-10. **Duplex-to-simplex transition**
(scheduling offset = 3)
(Storing duplex paper path, minimum duplex offset = 9)



$H_{i,j}$ $P_{i,j}$ $R_{i,j}$ ::= "Hint, Print, Request" triplet for sheet i, plate j.

j ::= PlateNumber

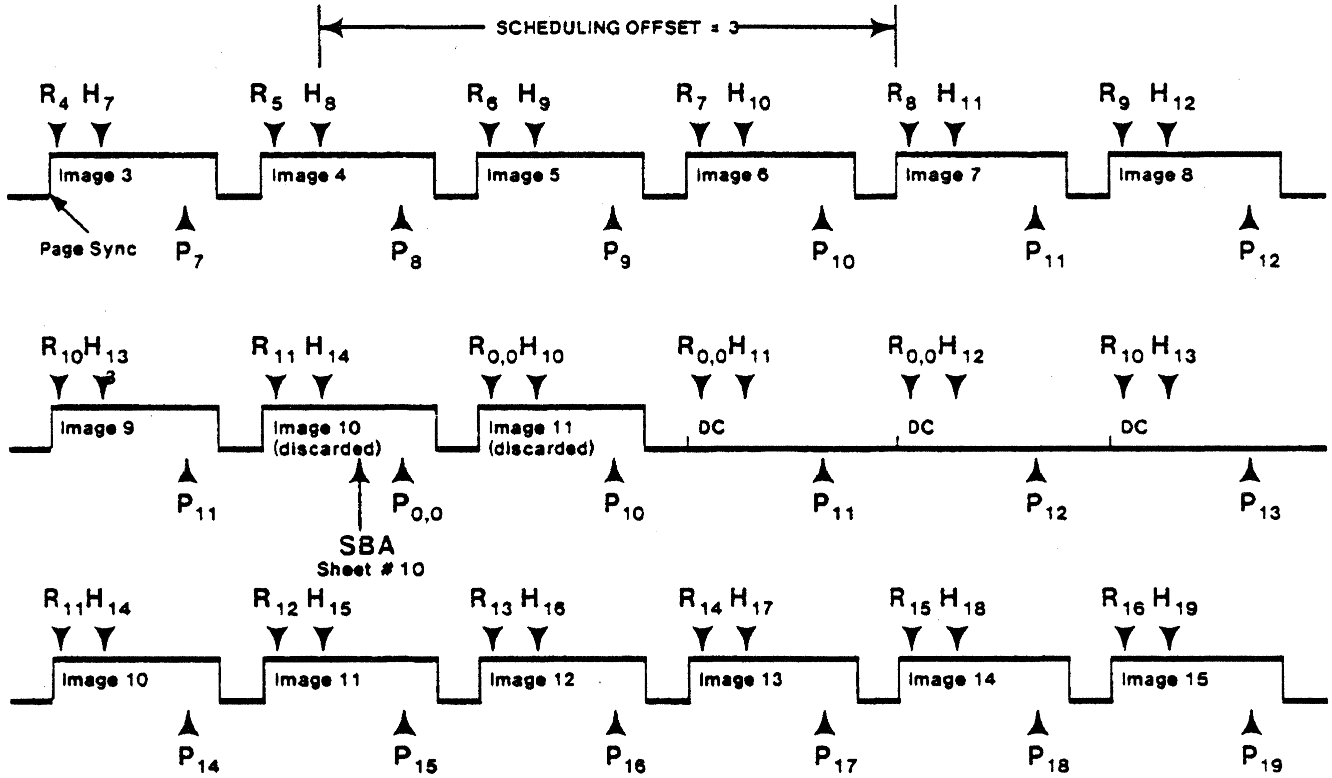
PlateNumber ::= simplex side | duplex side

simplex side ::= 1

duplex side ::= 2

NBR ::= NextBankRequest

Figure B6-11. **Simplex abort sequence** (scheduling offset = 3)



$H_{0,0}$ ::= "Hint" Dead Cycle (DC)

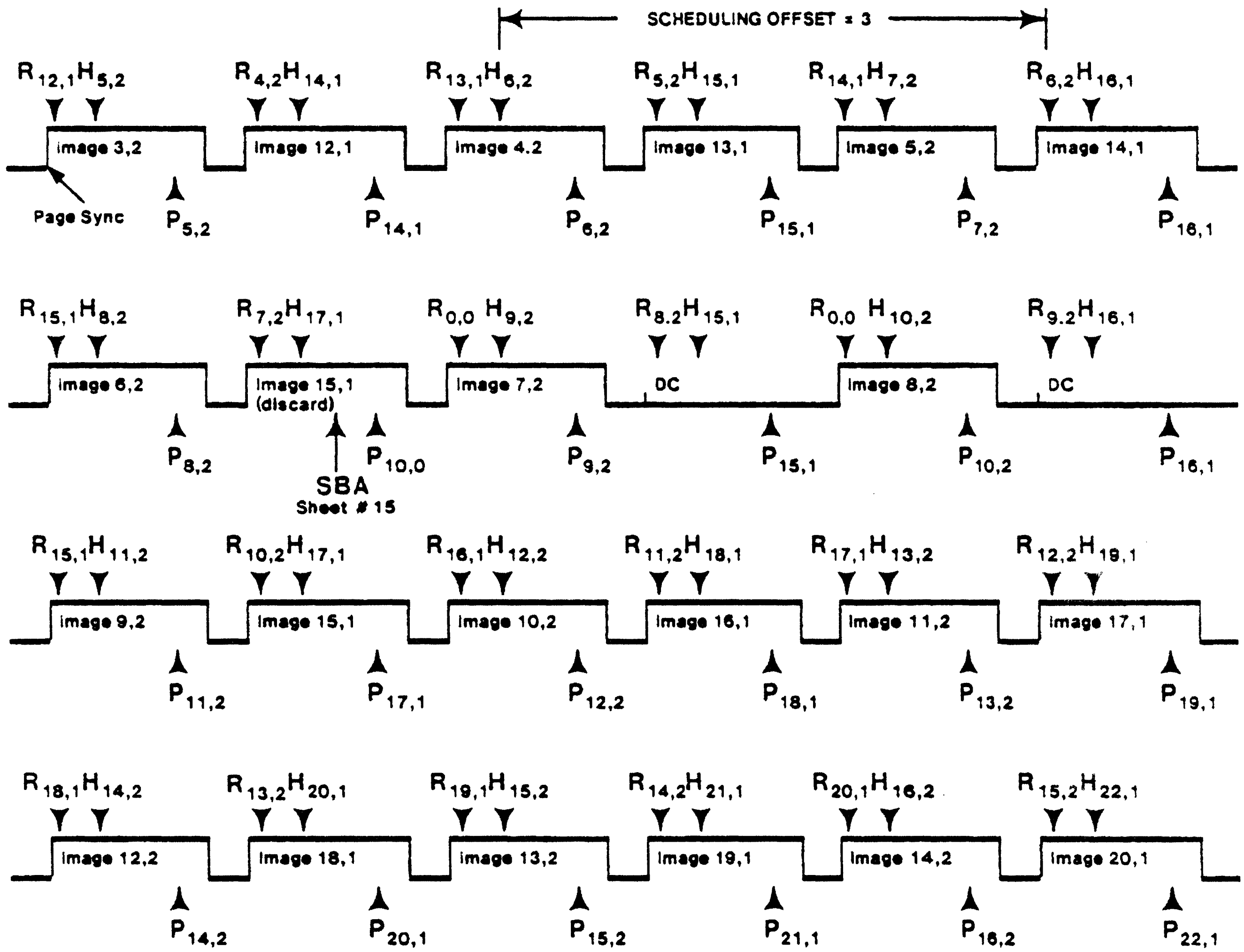
$P_{0,0}$ ::= Print Dead Cycle

$R_{0,0}$ ::= Request for a Dead Cycle

$H_n\ P_n\ R_n$ ::= n-th "Hint, Print, Request" triplet. (Only sheet nos. are shown.)

SBA ::= SheetBankAbort

Figure B6-12. **Duplex abort sequence, sheet abort during simplex side imaging**
(scheduling offset = 3)
(Storing duplex paper path, minimum duplex offset = 9)



$H_{i,j}$ $P_{i,j}$ $R_{i,j}$ :: = "Hint, Print, Request" triplet for sheet i, plate j.

j :: = PlateNumber

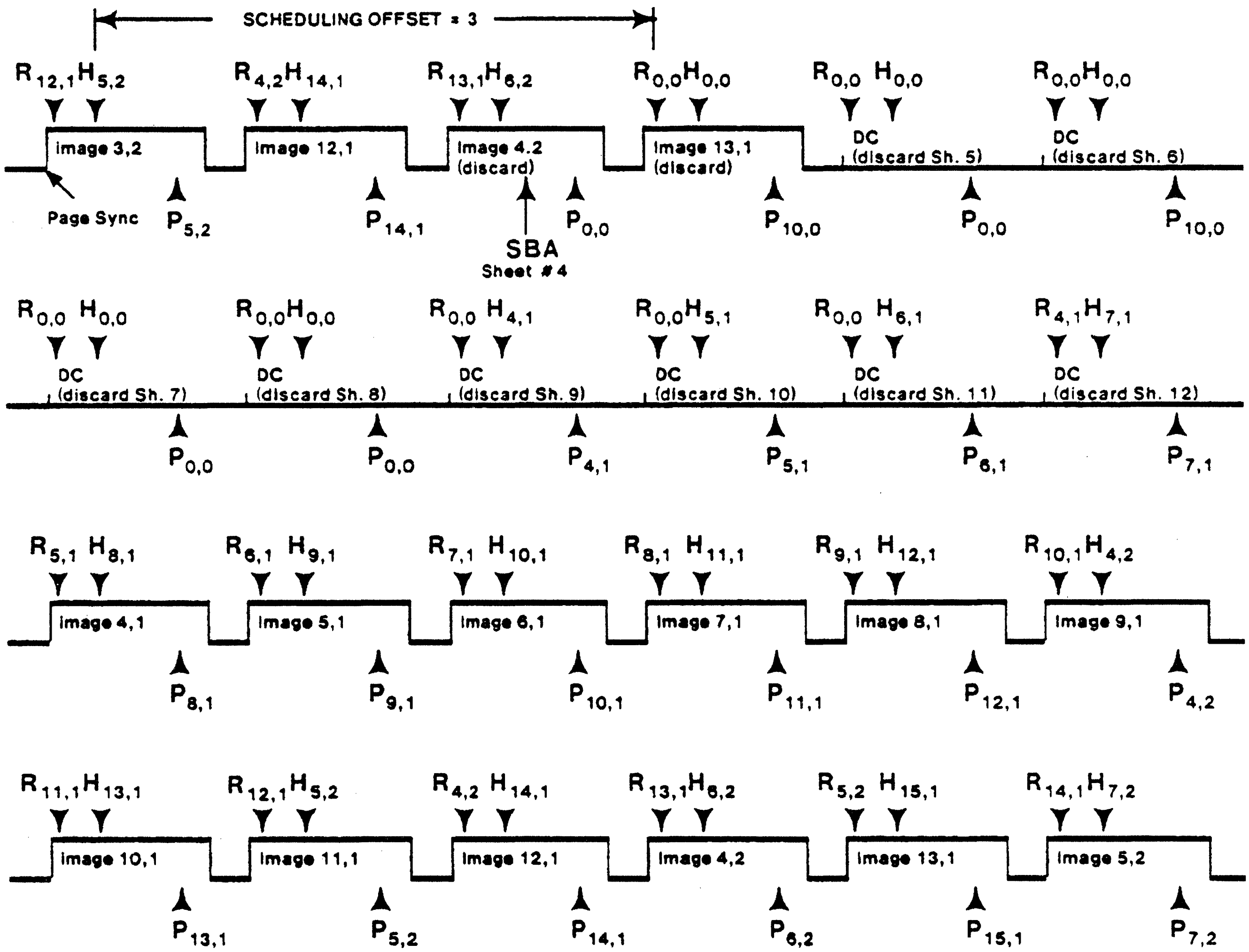
PlateNumber :: = simplex side | duplex side

simplex side :: = 1

duplex side :: = 2

SBA :: = SheetBankAbort

Figure B6-13. **Duplex abort sequence,**
sheet abort during duplex side imaging
(scheduling offset = 3)
(Storing duplex paper path, minimum duplex offset = 9)



$H_{i,j}$ $P_{i,j}$ $R_{i,j}$ ::= "Hint, Print, Request" triplet for sheet i, plate j.

j ::= PlateNumber

PlateNumber ::= simplex side | duplex side

simplex side ::= 1

duplex side ::= 2

SBA ::= SheetBankAbort

C. Operating sequence examples, bindexer destination

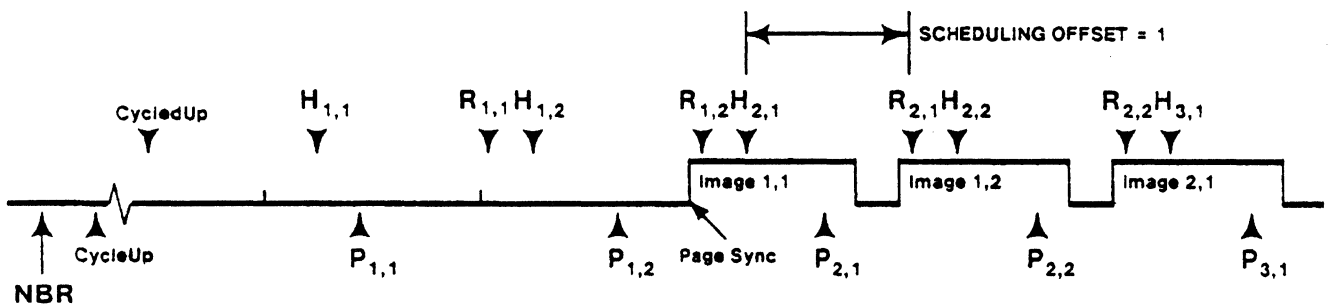
Following are three operating sequence examples which supplement those shown in section 6.5 and appendix B. The assumptions are:

a) Two collated copies to a bindexer,

b) Scheduling Offset = 1,

c) Racetrack duplex paper path, DuplexOffset = 8.

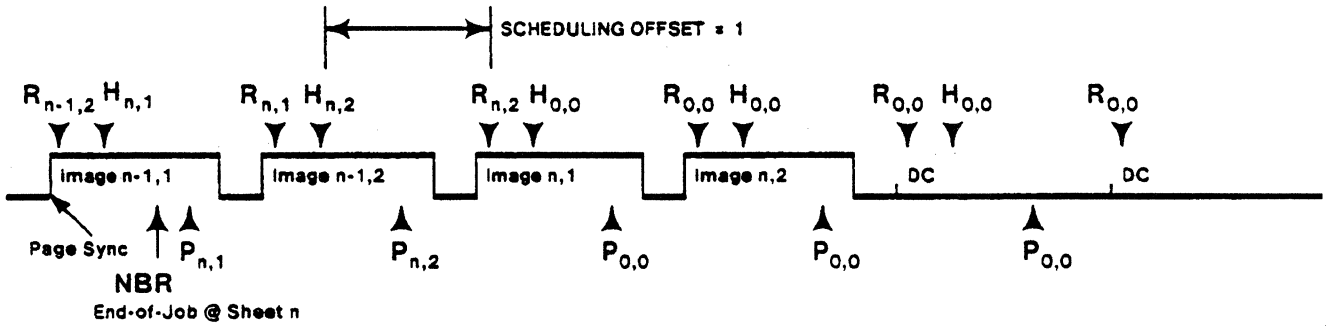
Figure C6-7-a. **Simplex cycle-up sequence**
(2 copies/bindexer destination)



$R_{0,0}$ ::= **Request for a Dead Cycle (DC)**

$H_{n,c}$ $P_{n,c}$ $R_{n,c}$ ::= **n-th "Hint, Print, Request" triplet. (c = CopyNumber)**

Figure C6-7-b. **Simplex cycle-down sequence**
(2 copies/bindexer destination)



$H_{0,0}$ ::= "Hint" Dead Cycle (DC)

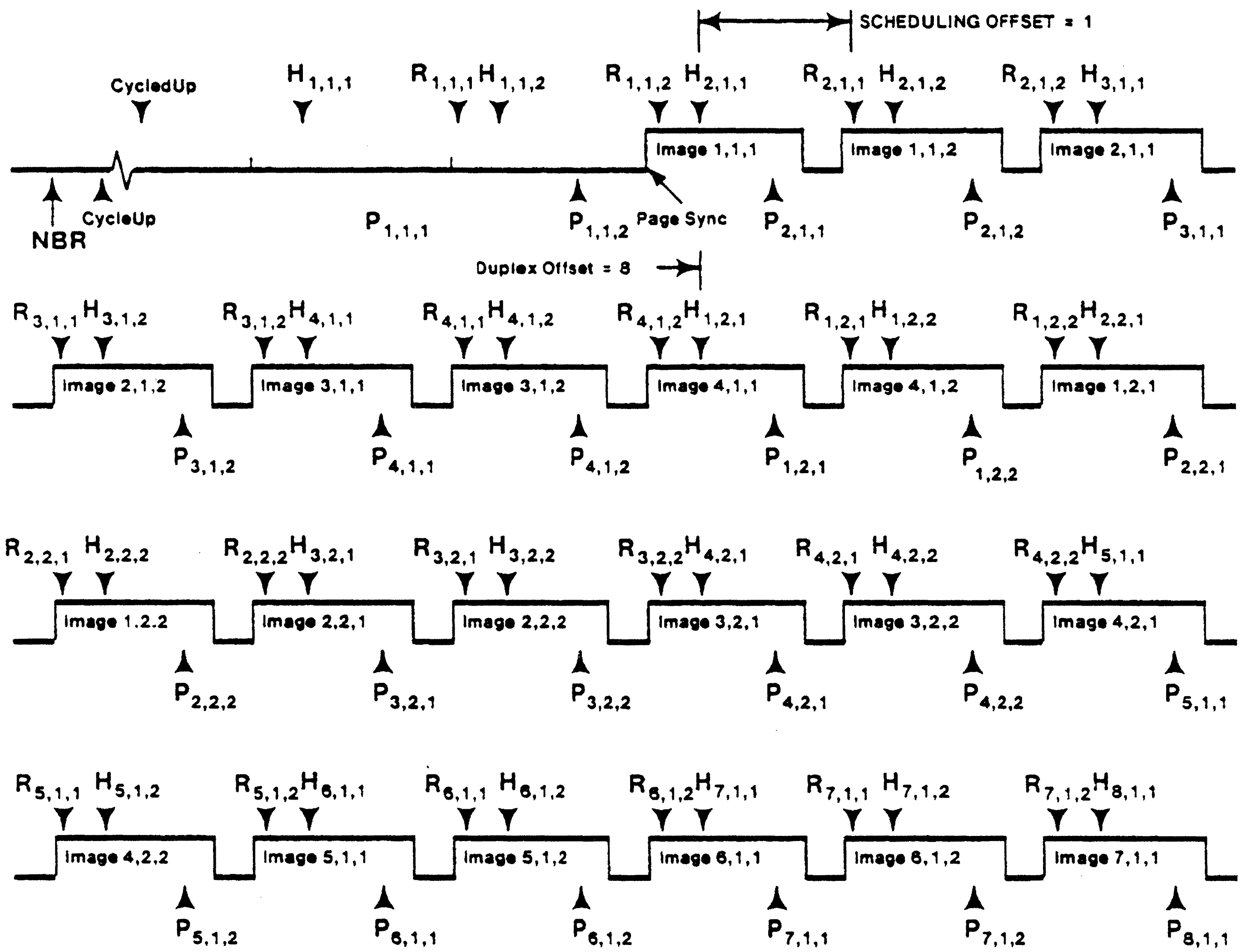
$P_{0,0}$ ::= Print Dead Cycle

$R_{0,0}$ ::= Request for a Dead Cycle

$H_{n,c}$ $P_{n,c}$ $R_{n,c}$ ::= n-th "Hint, Print, Request" triplet. (c = CopyNumber)

NBR ::= NextBankRequest

Figure C6-8. **Duplex cycle-up sequence (2 copies/bindexer destination)**
(scheduling offset =1)
(Race Track duplex paper path, duplex offset = 8)



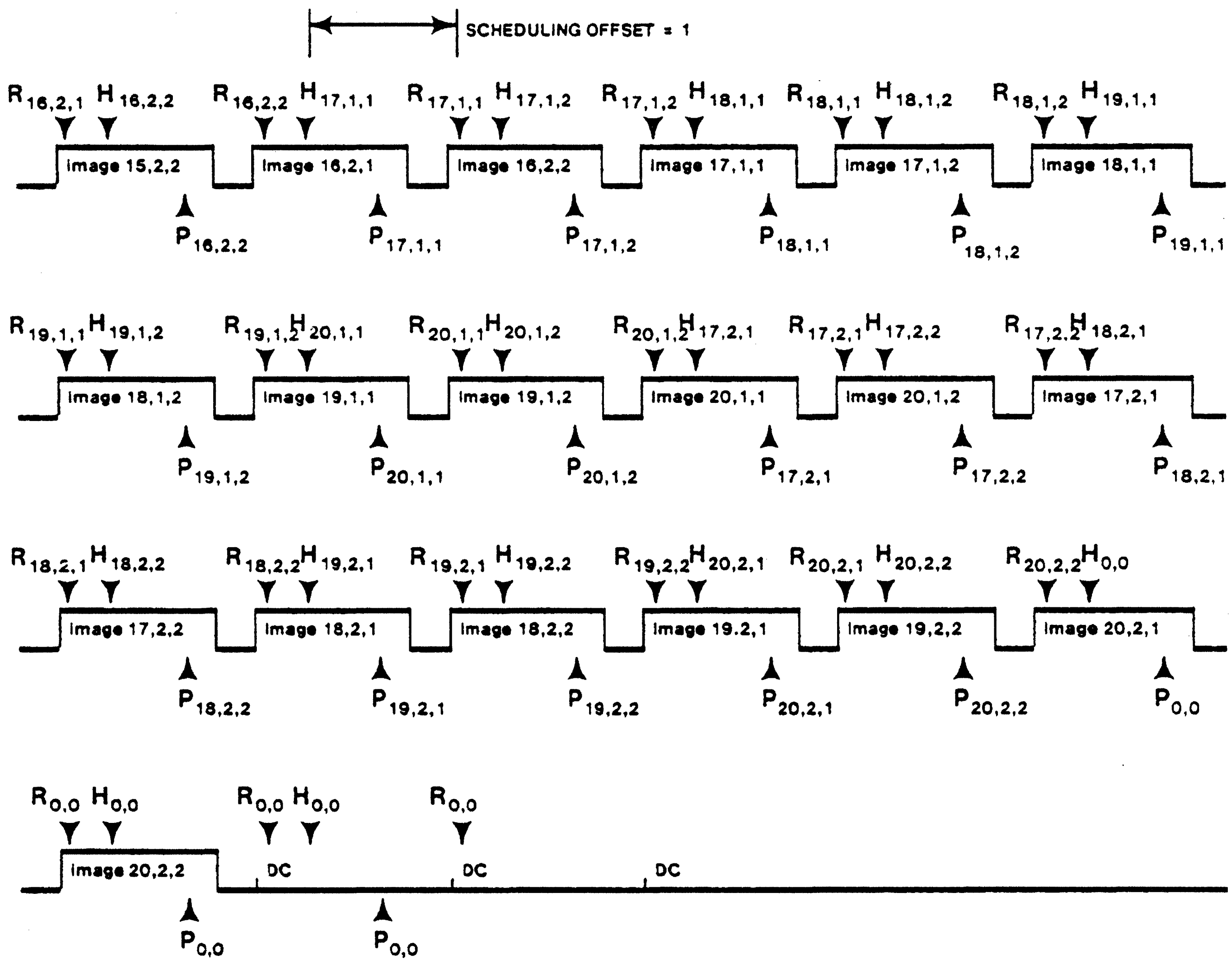
$H_{i,j,c}$ $P_{i,j,c}$ $R_{i,j,c}$ **::= "Hint, Print, Request" triplet for sheet i, plate j, copy c.**

i ::= SheetNumber
j ::= PlateNumber
c ::= CopyNumber

PlateNumber ::= simplex side | duplex side

simplex side ::= 1
duplex side ::= 2

Figure C6-9. **Duplex cycle-down sequence (2 copies/bindexer destination)**
(scheduling offset =1)
(Race Track duplex paper path, duplex offset = 8)



$H_{i,j,c}$ $P_{i,j,c}$ $R_{i,j,c}$ ::= "Hint, Print, Request" triplet for sheet i, plate j, copy c.

i ::= SheetNumber
j ::= PlateNumber
c ::= CopyNumber

PlateNumber ::= simplex side | duplex side

simplex side ::= 1
duplex side ::= 2

D. Recommended circuitry for power control interface

PSP control of IOT AC power is specified in section 8.4. The circuit shown in figure D-1 is recommended to implement this function.

The control actuator in the IOT may be a solid state or conventional relay of not less than 120 ohms. R4 must limit the short circuit current, **power control +** to ground, but allow nominal 50 ma load current. The worst case conditions for the latter are: power supply at lower limit of 11.4V, voltage drop across sinking transistor Q1 maximum of 0.4V, and R4 at its high tolerance end-of-life value, designated R4'. Under these conditions, R4' = 100 ohms, and R4 = 100/1.09 =91.7 ohms. For power dissipation reasons, it is recommended that R4 be composed of six 560 ohm 0.5 watt resistors in parallel, which provide an effective resistance of 93.33 ohms and effective power rating of 3 watts. With this value, the lowest current in a 120 ohm load is 11V/(120 + 1.09(93.33)) = 49.6 ma, the worst case short circuit current with R4 at low-tolerance, end-of-life value is 12.6V/(.91(93.33)) = 148.4 ma, and the corresponding worst-case power dissipation in R4 is 1.87 watts. Note that the effective power rating of R4 represents a derating of approximately 62 percent. This is less conservative than the usual 50 percent derating for continuous operation, but is sufficient to prevent damage to the resistors or to a printed wire board (max. allowable surface temperature = 122 degrees C) during this abnormal condition. A more conservative derating can be obtained by using eight 750 ohm 0.5 watt resistors, if desired. The highest current which the power source must supply to a 120 ohm load occurs with the power supply at its high limit of 12.6V, the voltage drop across Q1 at its minimum of 0.2V, and R4 at its minimum of .91(93.33) ohms. Under these conditions, the current is 60.5 ma. Under the same conditions, the highest voltage across a 50 ma load will be 8.15V.

R5 is to be selected so as to meet the operating requirements of the chosen control actuator, and is to be considered part of the load. For example, consider a nominal 6V relay with a 180 ohm ±10% coil, and a pull-in requirement of 4.8V. With a voltage pull-in requirement, the worst case which must be considered is minimum supply voltage (11.4V), maximum voltage drop across Q1 (0.4V), maximum end-of-life resistance of R4 (101.7 ohms), minimum coil resistance (162 ohms), and maximum end-of-life resistance of R5 = R5'. The current required in the 162 ohm coil to produce 4.8V is 29.6 ma. Under these conditions, R5 = R5'/1.09 = 98.7 ohms. To guarantee 4.8V across the coil, the next *lower* standard value is chosen, 91 ohms. With this value at high end of life, 99.2 ohms, the guaranteed minimum coil voltage is 4.91V. The worst-case power dissipation in R5 occurs with maximum supply voltage of 12.6V, minimum drop across Q1 of 0.2V, minimum value for R4 of 84.9 ohms, minimum coil resistance of 162 ohms, and maximum value for R5 of 99.2 ohms. It is 0.127 watts.

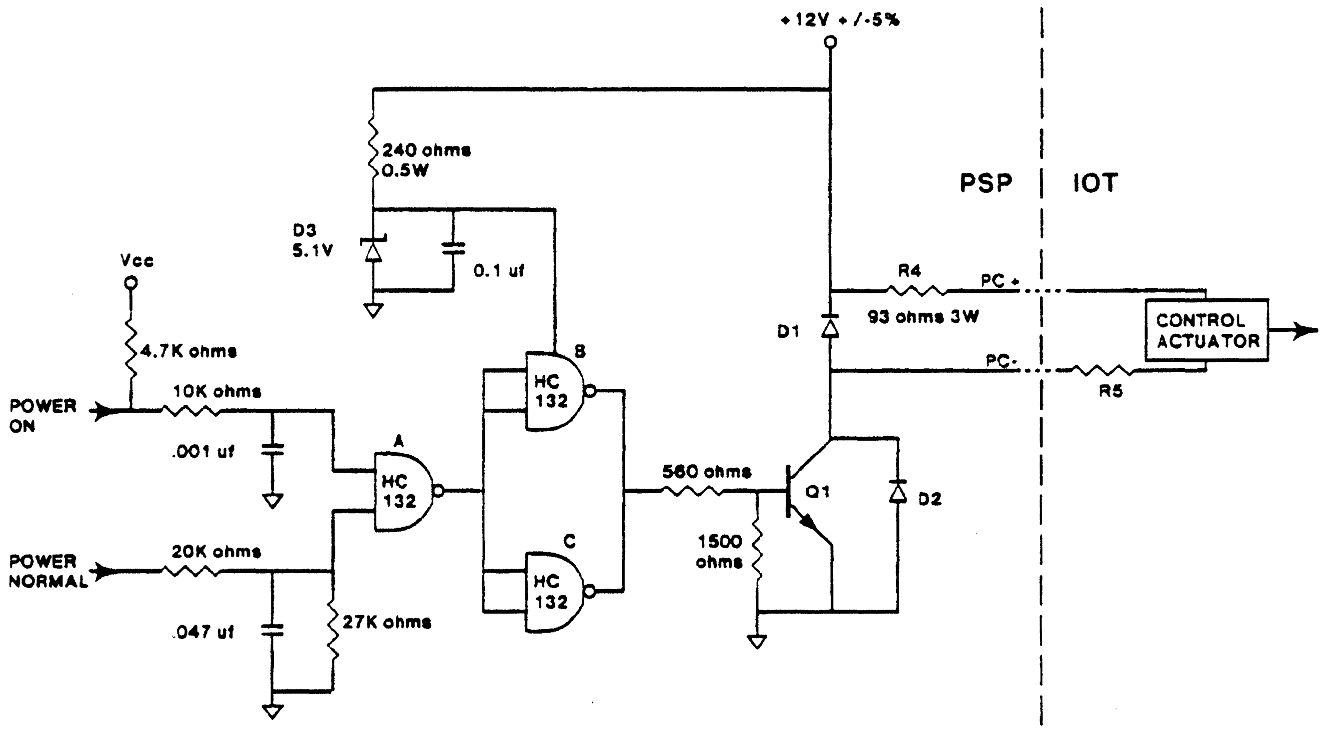
Consideration must also be given to the maximum power rating of the relay. The maximum operating current for this example is with maximum supply voltage of 12.6V, minimum drop across Q1 of 0.2V, minimum values for R4 and R5 of 84.9 and 82.8 ohms, respectively, and minimum coil resistance of 162 ohms. Under these conditions, the operating current is 37.6 ma, and the power dissipation in the coil is 0.229 watts. The maximum operating voltage across the coil occurs with all conditions the same, except coil resistance = 198 ohms. Under these conditions, the coil voltage is 6.7V and the power dissipation in the coil is 0.228 watts. These results would have to be checked against the power rating of the relay.

The active control circuitry has been designed with the following strategy. The +5.1V power for parallel gates B and C, which drive the output transistor Q1, is derived from the +12V source and will, thus, be into stable regulation following main power turn-on in the PSP, prior to both the +12V supply and to the +5V Vcc supply. The Power Normal signal is designed to remain low until the Vcc supply is into stable regulation. Thus, **power control +** is guaranteed to remain firmly in the off state during main power turn-on, and all elements of the control circuitry are guaranteed to be stable before the Power On signal can be effective at the interface. Chattering of the power actuator in the IOT, due to transient power supply anomalies in the PSP, is thereby avoided. The sequence is reversed during main power turn-off. That is, Power Normal goes low prior to Vcc leaving stable regulation, and thus also prior to +12V and +5.1V leaving stable regulation. Thus, **power control +** is guaranteed to be firmly in the off state during main power turn-off.

Diodes D1 and D2 prevent transient noise as great as 20 volts on the interface leads from changing either normal logic state of parallel driving gates B and C. Zener diode D3 and associated components assure that Q1 is controlled by gates B and C, and that the control is glitch free. These gates operate on supply voltage as low as 2V, and thus maintain control longer than any other circuits during the main power on and off transition

RC filters at the inputs to NAND gate A remove noise on the Power On and Power Normal signal lines. However, *once the Power Normal signal goes high, any glitch or noise greater than the logic threshold and longer than 10 microseconds on the Power On line can change the output state.*

Figure D-1. **Recommended circuitry for power control interface**



Part Numbers:

D1 707W01987
D2 707W01987
D3 707W00160
Q1 707W00081
Gates 733W03033

E. Alternate imaging modes

The primary imaging mode addressed by this specification is single color, 1 bit per pixel. (This includes full color when the individual color plates are transferred sequentially.) Alternate imaging modes such as trilevel highlight color (2 bits per pixel) and 2^n level grey scale, where n is integer 2 to 8, can also be accommodated. It is only necessary to follow prescribed signal formats at the interface.

Single-color (or full-color, sequential plate) is multiplexed onto the eight data paths of the parallel video data interface as specified in § 3.1.3.1, i.e., earliest bit of a scan line octet on Video Data 7, latest bit of an octet on Video Data 0. The alternate imaging modes are to be handled as follows and are summarized in figure E-1.

Trilevel Highlight Color (2 bits per pixel) shall be transferred four pixels at a time via the eight data paths of the parallel video data interface. The LSBs (least significant bits) of the latest through earliest pixels shall be transmitted on Video Data 0, 2, 4, and 6, respectively, and the MSBs (most significant bits) of the latest through earliest pixels shall be transmitted on Video Data 1, 3, 5, and 7, respectively. This provides up to 100 Megapixels per second. This recommendation applies to all speeds, even though lower pixel rates could be accommodated with fewer signal paths. The objective here is to minimize the number of product configurations.

For grey scale, signals with 8, 4, and 2 bits per pixel will utilize the full video bandwidth when multiplexed onto the 8 video data lines of the interface. Signals with 3, 5, 6, and 7 bits per pixel will utilize less than the full video bandwidth.

256-level grey scale (8 bits per pixel) shall be transferred one pixel at a time via the eight data paths of the parallel video data interface. The least significant bit shall be transmitted on Video Data 0 and the most significant bit shall be transmitted on Video Data 7. This provides up to 25 Megapixels per second.

16-level grey scale (4 bits per pixel) shall be transferred two pixels at a time via the eight data paths of the parallel video data interface. The LSB of the later pixel of a pixel pair (dipixel) shall be transmitted on Video Data 0, the MSB of the later pixel on Video Data 3, the LSB of the earlier pixel of a dipixel on Video Data 4, and the MSB of the earlier pixel on Video Data 7. This provides up to 50 Megapixels per second.

4-level grey scale (2 bits per pixel) shall be transferred four pixels at a time via the eight data paths of the parallel video data interface. The LSB of the latest pixel of a 4-pixel nibble shall be transmitted on Video Data 0, the MSB of the latest pixel on Video Data 1, the LSB of the later pixel on Video Data 2, the MSB of the later pixel on Video Data 3, and so on, as shown in figure E-1-A. This provides up to 100 Megapixels per second.

8-level grey scale (3 bits per pixel) shall be transferred two pixels at a time via six of the eight data paths of the parallel video data interface. The LSB of the later pixel of a pixel pair (dipixel) shall be transmitted on Video Data 0, the MSB of the later pixel on Video Data 2, the LSB of the earlier pixel of a dipixel on Video Data 4, and the MSB of the earlier pixel on Video Data 6. Less than the full bandwidth of the interface is utilized, as shown in figure E-1-B. This provides up to 50 Megapixels per second.

2n-level grey scale, where n is 5, 6, or 7, shall be transmitted one pixel at a time. In each case, the least significant bit of each pixel shall be transmitted on Video Data 0 and the most significant bit shall be transmitted on Video Data n-1. Thus, less than the full bandwidth of the interface is utilized for up to 25 Megapixels per second in each case.

Table E-1. **Signal assignments for alternate imaging modes**

Generic path name	Video signal assignments & clock definition			
	Trilevel HLC	**Grey scale 8-bit, 256-level**	**Grey scale 4-bit, 16-level**	**Grey scale 2-bit, 4-level**
Video Data 0	B/W bit of latest pixel	LSBit of pixel	LSB of later pixel	LSB of latest pixel
Video Data 1	R/W bit of latest pixel	•	•	MSB of latest pixel
Video Data 2	B/W bit of later pixel	•	•	LSB of later pixel
Video Data 3	R/W bit of later pixel	•	MSB of later pixel	MSB of later pixel
Video Data 4	B/W bit of earlier pixel	•	LSB of earlier pixel	LSB of earlier pixel
Video Data 5	R/W bit of earlier pixel	•	•	MSB of earlier pixel
Video Data 6	B/W bit of earliest pixel	•	•	LSB of earliest pixel
Video Data 7	R/W bit of earliest pixel	MSBit of pixel	MSB of earlier pixel	MSB of earliest pixel
Video Clock	Nibble Clock	Pixel Clock	DiPixel Clock	Nibble Clock

A. Highlight color and binary grey scale

Generic path name	Video signal assignments & clock definition			
	Grey scale 7-bit, 128-level	**Grey scale 6-bit, 64-level**	**Grey scale 5-bit, 32-level**	**Grey scale 3-bit, 8-level**
Video Data 0	LSBit of pixel	LSBit of pixel	LSBit of pixel	LSB of later pixel
Video Data 1	•	•	•	•
Video Data 2	•	•	•	MSB of later pixel
Video Data 3	•	•	•	(not used)
Video Data 4	•	•	MSBit of pixel	LSB of earlier pixel
Video Data 5	•	MSBit of pixel	(not used)	•
Video Data 6	MSBit of pixel	(not used)	(not used)	MSB of earlier pixel
Video Data 7	(not used)	(not used)	(not used)	(not used)
Video Clock	Pixel Clock	Pixel Clock	Pixel Clock	Dipixel Clock

B. Non-binary grey scale

F. Change history

V2.0

The following is a brief description of the principal changes that have been made to the various sections of this document, subsequent to their initial piecemeal approval, and embodied in Version 2.0. Chapters 1, 2, 3, 4, 7, 8, 9, 10, and 11 were originally approved by the Print Standards Committee in December, 1984, and published as XSIS document number 218412. Chapter 5, Command/Status Communications was approved in May, 1985, and chapter 6 Client Layer Protocol and related appendices A and B (now B and C) were approved in November 1985.

General Reference to the Distributed Printing and Reprographic Systems (DPRS) architecture as the exclusive domain of this generic specification has been removed throughout the document.

Chapter 1 The single criterion for designating high-end printers, i.e., video data rate 10 Megabits per second or greater, has been qualified by also including the type of data link required to support the speed and functionality of the printer.

Chapter 2 Discussion of complex printing arrangements has been removed to new appendix E, and replaced by a reference to appendix E.

Chapter 3 Section 3.1.1. Added note to the effect that an IOT may implement more than one Standard Image Frame.

Section 3.1.2. Added references to related information on page sync in sections 6.4.2 and 6.4.4.

Section 3.2. The optional interface signals IOT Reset (formerly section 3.2.3) and Power Monitor (formerly section 3.2.5) have been deleted. To effect this change, the text and figures have been modified as follows:

> Section 3.2.3. This paragraph now contains an updated description of the Power Control interface signal and its function, plus references to related information in chapters 6 and 8.
>
> Section 3.2.4. Deleted, subject moved to section 3.2.3.
>
> Section 3.2.5. Deleted.
>
> Figures 3.2 and 3.3. Power Monitor/IOT Reset signal removed.

Section 3.3. Text has been rewritten and removed to new appendix E, and replaced by a reference to appendix E.

Chapter 4 Section 4.1. Added reference to Page Sync regimen and operating sequences in chapter 6.

Chapter 5 Section 5.2.1. Redirected reference to new appendix E.

Section 5.3.9.e. Added reference to related information in chapter 6. Note on possible Ack time incompatibility upgraded from temporary to permanent note.

Section 5.4.3. Added reference to related information in chapter 6.

Figure 5.7. Data Link Communication Sequence, updated for compatibility with chapter 6.

Chapter 6

Section 6.2, 6.3. All paper width and paper length parameters have been changed from one byte to two bytes. To effect this change, the text and format sketches listed below have been modified accordingly:

- Section 6.2.1. Command parameter summary for PspNextBankRequest.
- Section 6.2.2. Status parameter summary for IotConfiguration and IotOperationalInfo.
- Section 6.3.2. Under PspNextBankRequest:
  - parameter summary and transmission order sketch;
  - formatted definition of Destination paper dimensions;
  - application note explaining previous method of designating paper size.
- Section 6.3.3. Under IotConfiguration:
  - parameter summary and transmission order sketch;
  - formatted definition of TopTray, Stacker, MultibinSorter, Bindexer, and Feeder paper dimensions;
  - application notes for information types, "Destination" and "Feeder", and Feeder format sketch.
- Section 6.3.3. Under IotOperationalInfo:
  - parameter summary and transmission order sketch;
  - formatted definition of Feeder paper dimensions;
  - application note for information type, "Feeder", and Feeder format sketch.

Section 6.3.3. The 1-bit parameters in status message IotConfiguration indicating implementation or non-implementation of optional IOT Reset and Power Monitor interface signals have been deleted. (Refer to chapter 3 changes, above.)

Section 6.4, 6.5, Appendices A and B. The timing of client layer command PspNextBankRequest has been moved one page-time earlier than previously prescribed. To effect this change, the text and figures have been modified as follows:

Section 6.4.2. The last two sentences of this paragraph have been changed to describe the new timing relationship.

Figure 6.4. The index of NBR has been changed from (n+x) to (n+1+x).

Figure 6.7-b. NBR(end-of-job @ sheet n) has been moved two page-times to the left (this was previously in error by one page-time too far to the right).

Section 6.5.2. The sixth sentence has been changed to reflect the change in Figure 6.7-b.

Figure 6.9. NBR(end-of-job @ sheet 19) has been moved one page-time earlier.

Section 6.5.4. Second sentence has been changed to reflect the changes in Figure 6.9.

Figure B6.7-b (formerly A6.7-b). NBR(end-of-job @ sheet n) has been moved one page-time to the left.

Figure B6.9 (formerly A6.9). NBR(end-of-job @ sheet 19) has been moved one page-time to the left.

Figure C6.7-b (formerly B6.7-b). NBR(end-of-job @ sheet n) has been moved two page-times to the left, i.e., just preceding P(n,1) (this was previously in error, one page-time too far to the right).

Chapter 7

Section 7.2. The timing analysis has been redone using cable specifications from the actual issued drawings (results are slightly more favorable).

Figure 7.3. Redrawn to agree with new timing analysis (see section 7.2, above.)

Chapter 8

Section 8.1. Extent of the timing re-analysis recommendation has been limited to changes in video or return clock circuits.

Section 8.2. Extent of the timing re-analysis recommendation has been limited to changes in video or return clock circuits.

Section 8.3. Note about exception conditions added at end of paragraph.

Section 8.4. Rewritten to reflect detailed circuit design for the Power control function now given in new appendix C. References to related information in chapter 6 added.

Section 8.5. Text on Power Monitor/IOT Reset function deleted (see section 3.2 changes, above) and replaced with text on Detection of Cable and Power Status at the Interface.

Section 8.7. Added note that each cable now has a spare pair (result of dropping Power Monitor/IOT Reset function).

Figure 8.5. Redrawn to reflect new Power Control circuit and deletion of Power Monitor/IOT Reset function.

Figure 8.6. New figure showing Circuits for Detection of Cable and Power Status at Interface. Old Figure 8.6 relabled Figure 8.7.

Figure 8.7. Connector symbols turned around to reflect actual construction. Interface demarcation lines identified. Old Figure 8.6.

Chapter 9

Section 9.1. Gender of PWB-mounted and cable-mounted connectors interchanged to reflect final design.

Section 9.2.1. All Xerox part numbers are now included (vendor's part numbers deleted).

Section 9.2.2. All Xerox part numbers are now included (vendor's part numbers deleted).

Section 9.3. Power Monitor pin assignments changed to "spare" (see section 3.2 changes, above). Reference to complex printing arrangements deleted.

Section 9.4.1. Reference to complex printing arrangements deleted.

Section 9.4.2. Power Monitor pin assignments changed to "spare" (see section 3.2 changes, above).

Section 9.4.3. Gender of PWB-mounted and cable-mounted connectors interchanged to reflect final design. Dimensions of bulkhead holes added.

Chapter 10

Section 10.1. The IEC Standard citation has been changed from IEC 435 to IEC 950. This standard has recently replaced both IEC 435 and IEC 380. Note that this paragraph is only advisory and, as such, does not state a requirement.

Section 10.2 Corrected Agency designation in last standard reference to "EEC." A note has been added concerning new emission standard, CISPR 22. Note that this paragraph is only advisory and, as such, does not state a requirement.

Appendix A (formerly chapter 11)

All SDLC references replaced with HDLC references. Character code reference added.

Appendix B (formerly appendix A)

Refer to figures B6.7-b and B6.9 (formerly figures A6-b and A6.9) under chapter 6 changes, above.

Appendix C (formerly appendix B)

Refer to figure C6.7-b (formerly figure B6-7-b) under chapter 6 changes, above.

Appendix D (formerly appendix C in January 20, 1986 revision)

New text and new Figure D.1 on Recommended Circuitry for Power Control Function. Corrected value of resistor in base of Q1 from 1K ohms (in January 20, 1986 revision) to 1.5K ohms.

Appendix E

Text and figures on Complex Printing Arrangements (from old section 3.3) rewritten with newer information. This appendix now contains all discussion of this subject in this document. (The contents of this appendix are deleted in the OEM version of this document.)

Appendix F

(Added.)

Glossary

(Added, formerly appendix D in January 20, 1986 revision.)

V2.1

The following is a brief description of the principal changes that have been made to Version 2.0 of this document, and embodied in Version 2.1.

Chapter 6

Section 6.3.2. Under PspNextBankRequest, bits d0 and d1 of the FutureFinishingOptions field have been assigned.

V2.2

A number of changes to Version 2.1 were published on an interim basis as Addendum 1 (XNSGS 108810a, November 1988). All of these changes are now included in Version 2.2. Addendum 1 is therefore rendered obsolete by Version 2.2. A number of additional changes have also been included in Version 2.2. All of the changes were requested by implementors. The following is a brief description of all the changes to Version 2.1 which are embodied in Version 2.2.

Chapter 6

New resolution parameter:

Provision has been made in the command/status message structure to specify resolution from sheet to sheet and from front to back of the same sheet. Two IOT specific resolutions are available for each side. The commands affected are PspNextBankRequest and PspPrint. The status messages affected are IotVideoHint and IotVideoRequest. This change supercedes the change published in Addendum 1 which provided four IOT specific resolutions without simplex/duplex designations.

BinFillLimit:

This parameter of the PspNextBankRequest command and its associated application note have been modified to extend its applicability to different output devices and the units of measurement have been made IOT specific.

InterruptJobASAP:

The application note for this command function of PspNextBankRequest has been modified to allow alternatives to the preferred action of delivering an interrupting job to a different destination than the interrupted job. Section 6.4.4, "Job interrupt protocol," has also been rewritten to be consistent with these changes.

Diagnostic utility numbering:

The PspRequestIotDiagnositc command and IotDiagnosticResponse status message have been restructed to expand the utility field from one byte to two byes, in accordance with the new Multi-National Standard for Diagnostic Program Numbers and Status Codes.

Finisher action:

Under the PspRequestIotAction command, definition of the Action field when device is FINISHER has been expanded to provide the capability to turn the finisher off by command, to set a time-out interval to be used by the finisher to remove power from the finisher binder, and to set binder tape length to accommodate both U.S. and non-U.S. paper length with fine adjustments for variances in paper length.

Numbering and sequencing conventions:

Section 6.4.3 has been rewritten to allow that an IOT may concurrently manage multiple jobs in every job category, *previous*, *current*, *next*, and *interrupted*. The IotConfiguration status message has been modified to report the maximum capability in only the two essential categories, interrupted jobs and the total of all jobs. The associated application note is modified accordingly. Section 6.4.4, "Job interrupt protocol," has also been rewritten to be consistent with these changes.

Machine Status and Substates:

Under the IotStateInfo status message, the application notes for machine state *CycledDownNotReady* and machine substates *FaultDetected* and *NonProductive* have been rewritten to eliminate an ambiguity which seemed to allow the *Productive* substate while in the *CycledDownNotReady* state.

Feeder and Destination parameters:

Under the IotOperationalInfo status message, the data fields for InformationTypes *FeederN* and InformationTypes *DestinationN* have been modified so that all the same attributes are reported for destinations as for feeders, and the feeder paper gauges and destination paper gauges have been made IOT specific.

Page Sync regimen:

Section 6.4.4 has been rewritten to require all IOTs to implement page sync so as to enable either a *continuous* page sync regimen or the previously defined regimen now referred to as the *discrete* regimen. This is to accommodate PSPs which accept only the continuous regimen or only the discrete regimen. Also, section 6.4.4 now allows page sync to appear at the interface under the discrete regimen when dead cycles that occur during job processing are used for sending test images to the IOT.

Stitch position:

The stitch position parameters in the NextBankTaskInfoC byte of the PspNextBankRequest command have been changed from an *offset* dimension to a *position* dimension and the measure has been made IOT specific.

Message reject:

The PspRejectIotMessage command and the IotRejectPspCommand status message have been restructured to include SheetNumber, CopyNumber, and JobNumber so as to uniquely identify which message is being rejected when these parameters are involved.

Contrast control:

Two new parameters, ContrastAction and ContrastData, have been added to the PspNextBankRequest command to give the PSP contract control capability.

Belt speed:

InformationType *BeltSpeed* and DataWord *SpeedInfo* have been added to the IotConfiguration status message to facilitate reporting belt speed.

Crash recovery protocol:

Section 6.4.7 on this subject has been rewritten to clarify which actions are performed by the data link layer and which are performed by the client layer.

Appendix E The discussion of complex printing arrangements in Version 2.1 has been replaced by specifications for handling alternate imaging modes, namely trilevel highlight color and grey scale. The OEM edition of Version 2.1, which was created by deleting the contents of appendix E, is now obsolete. There is no need for an OEM edition of Version 2.2.

Glossary

Following are the meanings of terms as used in this generic specification. They are not necessarily universal definitions.

background
See *Image background level.*

bank (parameter bank)
A set of parameters sent from the PSP to the IOT which describes how a series of images is to be processed with respect to paper source, output destination, finishing, color, simplex or duplex printing, etc.

bindexer
An IOT output device which accepts interleaved sheets of multiple copy-sets, collates the copy-sets simultaneously, and automatically ejects finished copy-sets to a stacker.

bit-map
The array of binary pixels which constitute an image, or one plate of a multiplate image. The array is in the form of a raster of sequential lines, each composed of a linear sequence of pixels.

Change Modification Entry (CME)
Changes to a plate or image which can be programmed to cause the otherwise identical plate or image to vary from copy to copy. Used, for example, to vary the information added to the several copies of a form to produce subsets of the master copy.

Control Interface
That part of the generic interface hardware and software which comprises the signalling channels for the Command, Status, and Power Control functions.

copy-set (copy)
The aggregate of sheets representing the complete set of sequential pages of a multi-page document as they appear at the output of the IOT. Copy-sets result at the output from collated delivery of single or multiple replications of multi-page documents. The abbreviated forms, *copy* or *set* may be used where there is no danger of ambiguity.

dead cycle
A cycle or pitch of the marking engine during which *page sync* is suppressed at the interface by the IOT and the PSP's Imaging subsystem delivers no video data to the interface. If paper is already commited to this cycle, it is discarded at the output.

duplex (printing)
Pertaining to a printing operation in which both sides of the paper are (or can be) imaged. Duplex printing is typically accomplished with two passes of a given sheet past a single imaging station, reversing sides between passes. (But see also Immediate Duplex and Simultaneous Duplex.) In a duplex printing operation, the *duplex side* of the paper is the back or reverse of the *simplex side*. See *simplex printing*.

duplex (transmission)
Pertaining to a point-to-point transmission system or facility in which information is sent in both directions simultaneously. Two-way simultaneous transmission.

edge marking
Placement of non-background portions of an image at the extreme edge(s) of the paper. Such marking is intended to be visible when the paper is stacked with other sheets. Because of the necessity to allow for paper registration error, edge marking typically results in imaging non-background information beyond the edge(s) of the paper.

end of scan edge With respect to the Standard Image Frame or to the paper, the edge which is perpendicular to the fast scan direction, and which is imaged last in those imaging systems which sequentially deliver pixels of image raster lines. In those imaging systems which employ line arrays which may image the complete line, or segments of the line, simultaneously, it is the edge corresponding to the end of the array which is typically loaded last with the video data.

fast-scan Applied to the direction or dimension which is parallel to the lines of the image raster, i.e. perpendicular to the direction of motion of the output medium (paper) during imaging.

gapless printer A printer whose IOT images continuously from one page sync true transition to the next, i.e. there is no time gap nor spatial gap between the trailing edge of one Standard Image Frame and the leading edge of the next Standard Image Frame. This is characteristic of web feed IOT's.

ground mecca Common point within an IOT, or within a PSP, where frame ground and green wire ground of the AC power source are connected together.

high-end Applied to a printer, an IOT, or a PSP which has a total video data rate greater than 10 Megabits per second at the PSP/IOT interface and whose speed or functionality requires the synchronous data link described in chapter 5.

high-light color (also hi-lite color) A printer capability providing one (or more) image color(s) in addition to the principal image color provided by the printer, for example, red in addition to black. The additional color(s) is intended for relatively sparse usage on a page i.e. for highlighting a limited number of characters, words, or other elements of the image. Highlighting may be invokable at the pixel, character, or word level, depending on the design of the machine. The high-light color is not necessarily applied with the same marking means nor the same marking method as used for the principal color. See also *trilevel hi-lite color*.

image (n.) The totality of information which is applied to one surface of an IOT's output medium. Note that an image may comprise more than one plate, and that the same image may be applied to the corresponding surfaces of more than one sheet in a multicopy job.

image (v.) To cause image information, e.g. a plate, to be applied to the intermediate transfer medium, or directly to the output medium, of an IOT.

image background level The video data signal level corresponding to the background level of a positive image, i.e. an image in which the variable data is rendered optically more dense than non-variable data on the output medium.

Imaging Interface That part of the generic interface hardware which comprises the twelve channels for page sync, line sync, byte clock, return byte clock, and eight video data signals in the parallel mode, or the five channels for page sync, line sync, pixel clock, return pixel clock, and one video data signal in the serial mode.

immediate duplex (also simultaneous duplex) A printing process in which both surfaces of a sheet of paper are imaged in a single pass past two separate imaging stations positioned to image opposite sides of the paper. The imaging stations may be directly opposed, in which case the imaging is truly simultaneous, or they may be offset along the paper path.

IOT (Image Output Terminal) (Defined in chapter 1)

job At the PSP input; a single Interpress Master file (or equivalent). At the IOT Interface and within the IOT; the aggregate of parameter banks which describe the processing of all sheets of all copy-sets which are to be printed from a single Interpress Master (or equivalent). Within the IOT, jobs are categorized as *previous*, *current*, *next*, or *interrupted*. (Refer to section 6.4.3).

leading edge With respect to the Standard Image Frame or to the paper, the edge perpendicular to the motion of the Standard Image Frame or paper which reaches the transfer station first, or in a direct imaging system, reaches the imaging station first.

long edge feed Cut-sheet paper feeding system in which the longer edge of the paper is perpendicular to the motion of the paper, and the fast scan direction is thus parallel to the longer edge.

marking engine The major electro-mechanical sub-system of an IOT which converts electrical data to marks on paper.

mecca See *ground mecca*.

N-to-1 Characteristic of a job which indicates that the logical pages of a document are delivered to the output destination last page first, first page last. This implies that the output sheets are delivered face up.

octet A group of eight pixels which are contiguous in a scan line.

one-to-N (1-to-N) Characteristic of a job which indicates that the logical pages of a document are delivered to the output destination first page first, last page last. This implies that the output sheets are delivered face down.

page video That image information which is intended to appear on the output medium.

parameter bank See *Bank*.

peer IOT Defined with respect to a PSP, a peer IOT is one which closely matches the performance of the PSP with respect to maximum total video data rate, document throughput rate, and data link frame processing time.

pitch (machine pitch) The interval between successive recurrences of the same event in the imaging cycle of the marking engine. For example, the interval between two successive positive transitions of page sync. This particular interval is also known as page-time, at the interface. Refer to figure 4-1.

pixel Picture element. At the IOT interface, a binary bit representing full density or zero density of one color, typically black, of the picture element being represented. For grey scale imaging, pixels are composed of 2 to 8 bits representing the density of the pixel. The bits of multi-bit pixels are transferred in parallel across the interface. (Refer to appendix E.)

plate The image information which is applied in one imaging cycle to one surface of the output medium (paper). Note that more than one plate may be applied to the same surface, in sequential imaging cycles, to form a composite image, e.g., a color image in which different colors are applied via separate plates. The term plate is used when describing the rendering of a particular set, or subset, of imaging information, e.g., whether it is rendered on the simplex side or duplex side of the sheet, and in what color.

PSP (Print Service Processor) Major element of a printer which accepts image information from the larger system in a standard format and converts it to raster format and delivers it in real time to the IOT.

race track A duplex paper path which does not store sheets, and therefore has the characteristic that the Duplex Offset (see IotConfiguration, section 6.3.3.) remains constant as the path is filled. A Race Track may or may not be capable of loading and unloading the reversing station in a single page-time. If not, it will require that dead cycles be alternated with the simplex imaged sheets as the paper path is filled. (See also *storing (duplex paper path)*.)

ROS Raster Output Scanner. The electro-optical subsystem of a marking engine.

set See *copy-set.*

sheet A single piece of output medium (e.g. paper) which is the receptor of an image on one or both of its surfaces. Each image may comprise one or more plates. Note that when the output is derived from a continuous web, the individual sheets may not exist as separate entities until the web is cut. The web may be cut just prior to delivery of the sheets to an IOT output device, or it may be taken up on an output roll, removed, and cut off-line.

simplex (printing) Pertaining to a printing operation in which only one side of the paper is imaged. In a duplex printing operation, the simplex side of the paper is the obverse or front side when the sheet is viewed as part of a normal English document.

simplex (transmission) Pertaining to a point-to-point transmission system or facility in which information is sent in one direction only.

slow–scan Applied to the direction or dimension which is parallel to the motion of the paper during imaging, i.e. perpendicular to the lines of the image raster.

stack The aggregate of sheets bearing the same image (or images if duplex) as they appear at the output of the IOT. Stacks result at the output from multiple replications of a single sheet, or from uncollated delivery of multiple replications of a multi-sheet document.

standard image frame A rectangular image space whose sides are parallel to the slow scan and fast scan directions of the IOT and whose dimensions are equal to the dimensions of the largest size paper which an IOT can accommodate, plus registration tolerance in the respective directions. An IOT may be readjustable to implement more than one Standard Image Frame. Such readjustment may require the machine to be cycled down.

start of scan edge With respect to the Standard Image Frame or to the paper, the edge which is perpendicular to the fast scan direction, and which is imaged first in those imaging systems which deliver pixels of image raster lines in sequence. In those imaging systems which employ line arrays which may image the complete line, or segments of the line, simultaneously, it is the edge corresponding to the end of the array which is normally loaded first with the video data.

storing (duplex paper path) A duplex paper path which temporarily stores simplex imaged sheets so that alternate dead cycles are not required when the paper path is being filled. A storing duplex paper path has the characteristic that the DuplexOffset (see IotConfiguration, section 6.3.3.) increments, from the minimum to twice the minimum minus 1, as the paper path is filled. See also *race track*.

task Within the IOT, the processing represented by a single parameter bank.

total video data rate Number of bits per unit of time transferred across the imaging interface, whether serially or in parallel.

trailing edge With respect to the Standard Image Frame or to the paper, the edge perpendicular to the motion of the Standard Image Frame or paper which reaches the transfer station last, or in a direct imaging system, reaches the imaging station last.

trilevel hi-lite color An imaging technique in which the image for the principal color and the image for a single hi-lite color are delivered simultaneously to the intermediate imaging medium, and the image is developed and transfered in a single pass of the output medium.

web feed Characteristic of an IOT in which the output medium is supplied in a continuous web from a roll and is not cut into individual sheets until after imaging. The web may be cut just prior to delivery of the sheets to an IOT output device, or it may be re-rolled, removed, and cut off-line. The width of the web corresponds to the fast scan dimension, which is typically the length of the final cut-sheet.

Xerox Corporation
Xerox Systems Institute
475 Oakmead Parkway
Sunnyvale, CA 94086

Printed in U.S.A.