Semiconductors

AN10045

ISP1582/83 Clearing an IN Buffer

Rev. 01— 22 February 2005

Application note

Document information

Info	Content
Keywords	isp1582; isp1583; usb; universal serial bus; peripheral controller
Abstract	This document explains the clearing of an IN buffer in the ISP1582 and the ISP1583.

PHILIPS

Revision history

Rev	Date	Description
01	20050222	First release.

Contact information

For additional information, please visit: http://www.semiconductors.philips.com

For sales office addresses, please send an email to: sales.addresses@www.semiconductors.philips.com

1. Introduction

This document explains how an IN endpoint in the ISP1582 and ISP1583 can be cleared.

When the USB connector is pulled out during a data transfer, there may still be residual data in the IN endpoint buffer in use. This can cause data corruption when the next data transaction is scheduled. To avoid this, clear the IN endpoint buffer properly. A soft reset or a bus reset cannot properly clear the IN endpoint because the endpoint is like a state machine, and therefore, issuing a bus reset or soft reset does not clear the FIFO. A bus reset only resets some register settings but does not interfere with the state machine. The ISP1582/83 design is such that you need to issue a clear buffer command to clear the FIFO.

This is a general document to explain the clearing of an IN buffer in the ISP1582/83. A Mass Storage application is used as an example.

2. Mass storage application example

Fig 1 shows the block diagram of a Mass Storage application connection using the ISP1583 Mass Storage evaluation (eval) kit.

If you suddenly disconnect a USB cable in the midst of a file transfer between an ATA/ATAPI device, for example, a hard disk, to the host PC, chances are high that there will be residual data in the IN buffer. When the USB cable is reconnected, this residual data in the IN buffer from the previous connection will be sent to the host on the next IN token.

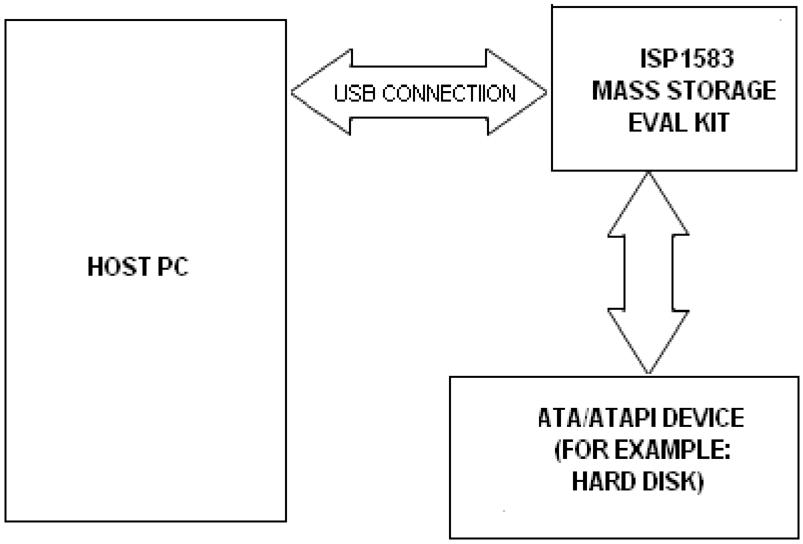
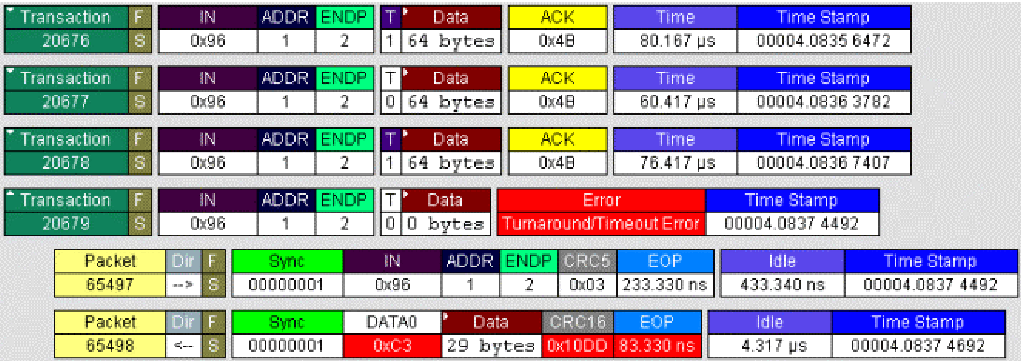


Fig 1. Block diagram showing the connection of the ISP1583 Mass Storage eval kit.

Fig 2 shows the CATC trace of the bus turnaround error because of the sudden disconnection of the USB cable while the file transfer from the hard disk to the host PC is in progress.



Transaction	F S	IN	ADDR	ENDP	T	Data	ACK	Time	Time Stamp
20676	S	0x96	1	2	1	64 bytes	0x4B	80.167 µs	00004.0835 6472

Transaction	F S	IN	ADDR	ENDP	T	Data	ACK	Time	Time Stamp
20677	S	0x96	1	2	0	64 bytes	0x4B	60.417 µs	00004.0836 3782

Transaction	F S	IN	ADDR	ENDP	T	Data	ACK	Time	Time Stamp
20678	S	0x96	1	2	1	64 bytes	0x4B	76.417 µs	00004.0836 7407

Transaction	F S	IN	ADDR	ENDP	T	Data	Error	Time Stamp
20679	S	0x96	1	2	0	0 bytes	Turnaround/Timeout Error	00004.0837 4492

Packet	Dir	F S	Sync	IN	ADDR	ENDP	CRC5	EOP	Idle	Time Stamp
65497	-->	S	00000001	0x96	1	2	0x03	233.330 ns	433.340 ns	00004.0837 4492

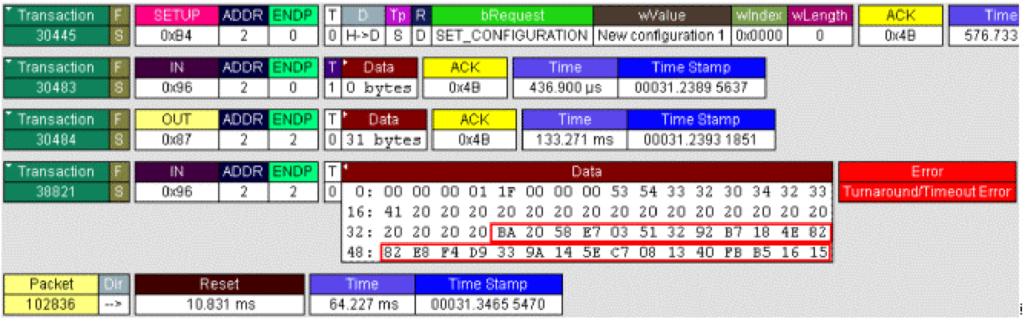
Packet	Dir	F S	Sync	DATA0	Data	CRC16	EOP	Idle	Time Stamp
65498	<--	S	00000001	0xC3	29 bytes	0x10DD	83.330 ns	4.317 µs	00004.0837 4692

Fig 2. CATC trace showing the bus turnaround error because of sudden disconnection.

After the USB cable is reconnected, the Mass Storage device will go through an enumeration process, during which the bus will be reset. After every bus reset, the firmware must reinitialize the initialization registers. It is found that after reconfiguring the endpoints, the IN buffer—that is, endpoint 2 IN—is still not cleared.

For endpoint 2 IN buffer, the firmware writes a short packet of 36 B. A full packet of 64 B, however, is validated and sent to the host; see Fig 3. The host, however, is expecting only 36 B of data and is confused when it receives a MaxPacketSize of 64 B. Therefore, the host resets.

Fig 3 shows the CATC trace of a full packet of 64 B sent to the host instead of a short packet that results in the bus turnaround error.



Transaction	F S	SETUP	ADDR	ENDP	T	D	Tp	R	bRequest	wValue	wIndex	wLength	ACK	Time
30445	S	0xB4	2	0	0	H->D	S	D	SET_CONFIGURATION	New configuration 1	0x0000	0	0x4B	576.733

Transaction	F S	IN	ADDR	ENDP	T	Data	ACK	Time	Time Stamp
30483	S	0x96	2	0	1	0 bytes	0x4B	436.900 µs	00031.2389 5637

Transaction	F S	OUT	ADDR	ENDP	T	Data	ACK	Time	Time Stamp
30484	S	0x87	2	2	0	31 bytes	0x4B	133.271 ms	00031.2393 1851

Transaction	F S	IN	ADDR	ENDP	T	Data	Error
38821	S	0x96	2	2	0	see dump below	Turnaround/Timeout Error

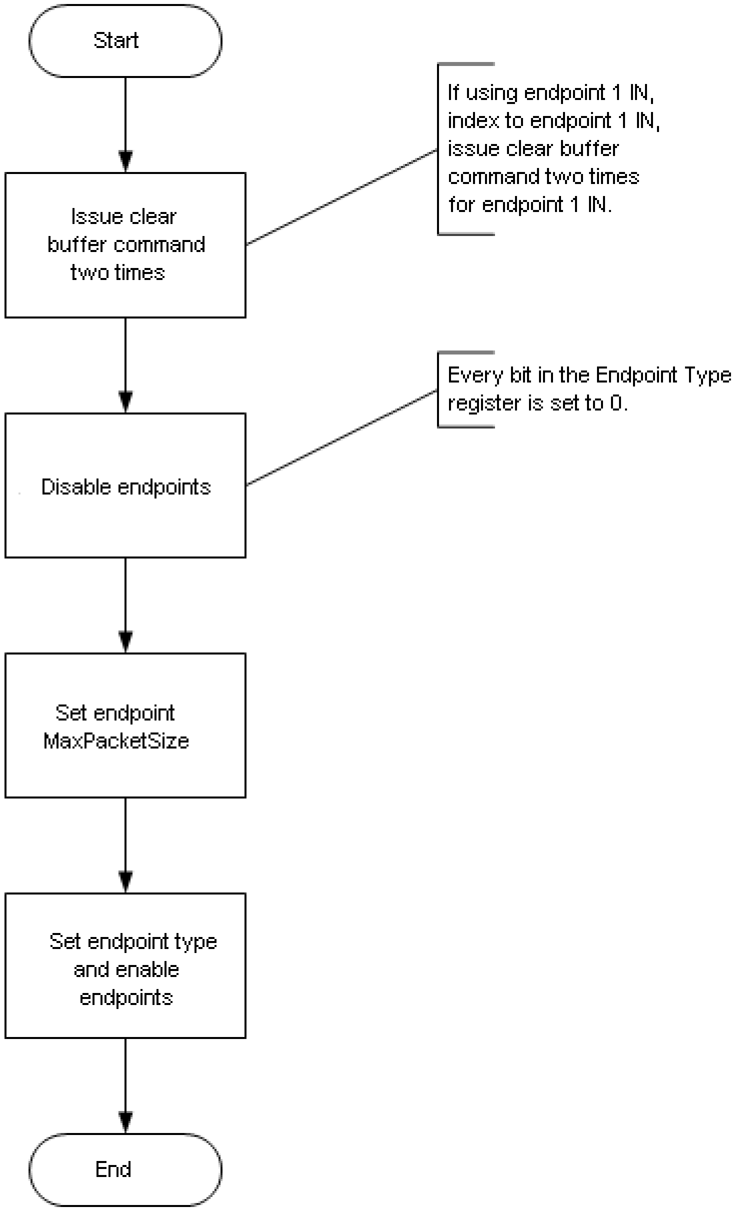
```
 0: 00 00 00 01 1F 00 00 00 53 54 33 32 30 34 32 33
16: 41 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
32: 20 20 20 20 BA 20 58 E7 03 51 32 92 B7 18 4E 82
48: 82 E8 F4 D9 33 9A 14 5E C7 08 13 40 FB B5 16 15
```

Packet	Dir	Reset	Time	Time Stamp
102836	-->	10.831 ms	64.227 ms	00031.3465 5470

Fig 3. CATC trace showing a full packet of 64 B sent to the host.

3. Clearing the IN buffer

Fig 4 shows the flowchart to properly clear the IN buffer. Use this flow to configure endpoints for the ISP1582/83. The firmware must issue two clear buffer commands, if double buffering is enabled, before configuring the size and type of endpoints. This will ensure proper clearing of the IN buffer.


Start
Issue clear buffer command two times
Disable endpoints
Set endpoint MaxPacketSize
Set endpoint type and enable endpoints
End

If using endpoint 1 IN, index to endpoint 1 IN, issue clear buffer command two times for endpoint 1 IN.

Every bit in the Endpoint Type register is set to 0.

Fig 4. Flowchart showing the correct way to clear the IN buffer.

4. Sample code on clearing of the IN buffer

The following is a sample code to clear the endpoint 2 IN buffer.

```
void Init_Endpoint(void)
{
   //Issue 2 Clear Buffer command for Endpoint 2 In
   D14_Cntrl_Reg.D14_DMA_ENDPOINT = 2;
      D14_Cntrl_Reg.D14_ENDPT_INDEX = 5;

   D14_Cntrl_Reg.D14_CONTROL_FUNCTION.BITS.CLBUF = 1;
   D14_Cntrl_Reg.D14_DMA_ENDPOINT = 2;
      D14_Cntrl_Reg.D14_ENDPT_INDEX = 5;

      D14_Cntrl_Reg.D14_CONTROL_FUNCTION.BITS.CLBUF = 1;

   //check if device in full speed state
   if(Kernel_Flag.BITS.HS_FS_State == FULL_SPEED)
   {
      //Bulk Out MaxPacketSize Endpoint
      D14_Cntrl_Reg.D14_ENDPT_INDEX = 4;
         D14_Cntrl_Reg.D14_ENDPT_MAXPKTSIZE.VALUE = 0x4000;

      //Bulk In MaxPacketSize Endpoint
      D14_Cntrl_Reg.D14_ENDPT_INDEX = 5;
         D14_Cntrl_Reg.D14_ENDPT_MAXPKTSIZE.VALUE = 0x4000;

      //Bulk Out Endpoint Type
      D14_Cntrl_Reg.D14_ENDPT_INDEX = 4;
         D14_Cntrl_Reg.D14_ENDPT_TYPE.VALUE = 0x1600;

//Bulk In Endpoint Type
      D14_Cntrl_Reg.D14_ENDPT_INDEX = 5;
         D14_Cntrl_Reg.D14_ENDPT_TYPE.VALUE = 0x1600;

//Enable FIFO
      D14_Cntrl_Reg.D14_ENDPT_INDEX = 4;
      D14_Cntrl_Reg.D14_ENDPT_TYPE.VALUE |= 0x0800;

      //Enable FIFO
      D14_Cntrl_Reg.D14_ENDPT_INDEX = 5;
      D14_Cntrl_Reg.D14_ENDPT_TYPE.VALUE |= 0x0800;
   }

      //check if device in high speed
   if(Kernel_Flag.BITS.HS_FS_State == HIGH_SPEED)
   {
      //Bulk Out MaxPacketSize Endpoint
      D14_Cntrl_Reg.D14_ENDPT_INDEX = 4;
         D14_Cntrl_Reg.D14_ENDPT_MAXPKTSIZE.VALUE = 0x0002;

      //Bulk In MaxPacketSize Endpoint
      D14_Cntrl_Reg.D14_ENDPT_INDEX = 5;
         D14_Cntrl_Reg.D14_ENDPT_MAXPKTSIZE.VALUE = 0x0002;

      //Bulk Out Endpoint Type
      D14_Cntrl_Reg.D14_ENDPT_INDEX = 4;
      D14_Cntrl_Reg.D14_ENDPT_TYPE.VALUE = 0x1600;

      //Bulk In Endpoint Type
      D14_Cntrl_Reg.D14_ENDPT_INDEX = 5;
         D14_Cntrl_Reg.D14_ENDPT_TYPE.VALUE = 0x1600;

      //Enable FIFO
      D14_Cntrl_Reg.D14_ENDPT_INDEX = 4;
      D14_Cntrl_Reg.D14_ENDPT_TYPE.VALUE |= 0x0800;

         //Enable FIFO
      D14_Cntrl_Reg.D14_ENDPT_INDEX = 5;
      D14_Cntrl_Reg.D14_ENDPT_TYPE.VALUE |= 0x0800;
   }
   //set to DMA endpoint 1
   //set to Endpoint Index to 2
   //this is to prevent the endpoint index pointing to the same endpoint
D14_Cntrl_Reg.D14_DMA_ENDPOINT = 2;
   D14_Cntrl_Reg.D14_ENDPT_INDEX = 5;

   D14_Cntrl_Reg.D14_CONTROL_FUNCTION.BITS.CLBUF = 1;

   D14_Cntrl_Reg.D14_DMA_ENDPOINT = 2;
      D14_Cntrl_Reg.D14_ENDPT_INDEX = 5;

      D14_Cntrl_Reg.D14_CONTROL_FUNCTION.BITS.CLBUF = 1;

      //Enable device and reset the device address
   D14_Cntrl_Reg.D14_ADDRESS.VALUE = 0x80;
}
```

5. Disclaimers

Life support — These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips Semiconductors customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips Semiconductors for any damages resulting from such application.

Right to make changes — Philips Semiconductors reserves the right to make changes in the products - including circuits, standard cells, and/or software - described or contained herein in order to improve design and/or performance. When the product is in full production (status 'Production'), relevant changes will be communicated via a Customer Product/Process Change Notification (CPCN). Philips Semiconductors assumes no responsibility or liability for the use of any of these products, conveys no licence or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified.

Application information — Applications that are described herein for any of these products are for illustrative purposes only. Philips Semiconductors make no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

6. Contents

Date of release: 22 February 2005

Published in The Netherlands