Semiconductors

AN10040

ISP1761 Peripheral DMA Initialization

Rev. 01 — 7 September 2004 Application note

Document information

Info	Content
Keywords	isp1761, usb, universal serial bus, dma
Abstract	This document explains the registers that are involved when using DMA on the peripheral controller of ISP1761.

PHILIPS

Revision history

Rev	Date	Description
01	20040907	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 application note explains the registers that are involved when using direct memory access (DMA) on the peripheral controller of ISP1761.

2. DMA initialization

The peripheral controller in the ISP1761 is a slave DMA device. It requires an external DMA master to acknowledge (ACK) a transfer.

To reduce power consumption, a controllable clock that drives the DMA controller circuits is turned off by default. If the DMA functionality is required by an application, DMACLKON (bit 9 of the Mode register) must be enabled during initialization of the peripheral controller, see Table 1:. If DMA is not required by an application, DMACLKON can be permanently disabled to save power. The burst counter, DMA bus width, and the polarity of DREQ and DACK must be set accordingly.

Table 1: Mode register (address: 020Ch)

Bit	15	14	13	12	11	10	9	8
Symbol	reserved(1)						DMACLKON	VBUSSTAT
Reset	-	-	-	-	-	-	0	-
Bus reset	-	-	-	-	-	-	0	-
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	CLKAON	SNDRSU	GOSUSP	SFRESET	GLINTENA	WKUPCS	reserved(1)	
Reset	0	0	0	0	0	0	0	0
Bus reset	0	0	0	0	Unchanged	0	0	Unchanged
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

(1) The reserved bits should always be written with the reset value.

Table 2: DMA Burst Counter register (address: 0264h)

Bit	15	14	13	12	11	10	9	8
Symbol	reserved(1)			BURSTCOUNTER[12:8]				
Reset	-	-	-	0	0	0	0	0
Bus reset	-	-	-	0	0	0	0	0
Access	-	-	-	R/W	R/W	R/W	R/W	R/W
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	BURSTCOUNTER[7:0]							
Reset	0	0	0	0	0	0	1	0
Bus reset	0	0	0	0	0	0	1	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

(1) The reserved bits should always be written with the reset value.

Although the peripheral controller in the ISP1761 is adapted from the ISP1582, the ISP1761 peripheral controller has by default only one GDMA handshake signal setting. The WR_N signal strobes data from the DMA bus onto the ISP1761 peripheral controller. The RD_N signal strobes data from the ISP1761 peripheral controller onto the DMA bus. In the ISP1582, DIOR and DIOW strobe data from and to the ISP1582, when DMA is in use; it also has a DACK-only mode. On the other hand, in the ISP1761 peripheral controller, DMA handshaking signals such as DACK-only, DIOR and DIOW are not available.

The ISP1582 supports two DMA modes: EOT and counter. The ISP1761 supports only counter mode. To enable counter mode, ensure that DIS_XFER_CNT in the DcDMAConfiguration register is set to zero, see Table 3:. ISP1761 does not support the external EOT mode; so, to make the EOT function invalid, set bit EOT_POL in the DMA Hardware register to logic 1; see Table 4:.

Set the DMA width according to the bus width.

Table 3: DcDMAConfiguration register (address: 0238h)

Bit	15	14	13	12	11	10	9	8
Symbol	reserved(1)							
Reset	0	0	0	0	0	0	0	0
Bus reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	DIS_XFER_CNT		reserved(1)		MODE[1:0]		reserved(1)	WIDTH
Reset	0	0	0	0	0	0	1	0
Bus reset	0	0	0	0	0	0	1	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

(1) The reserved bits should always be written with the reset value.

Table 4: DMA Hardware register (address: 023Ch)

Bit	7	6	5	4	3	2	1	0
Symbol	ENDIAN[1:0]		EOT_POL	reserved(1)	DACK_POL	DREQ_POL	reserved(1)	
Reset	0	0	0	0	0	1	0	0
Bus reset	0	0	0	0	0	1	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

(1) The reserved bits should always be written with the reset value.

Before starting the DMA transfer, preset the interrupt enable bit IEDMA in the DcInterruptEnable register and the DMA Interrupt Enable register. The ISP1761 supports two interrupt trigger modes: level and edge. The pulse width in edge mode is determined by setting the Interrupt Pulse Width register. The default value of the register is 1Eh, which indicates that the interrupt pulse width is 1 µs. The minimum interrupt pulse width is approximately 30 ns, when set to logic 1. Do not write a zero to this register. The interrupt polarity must also be correctly set.

Remark: The DMA can be performed on all endpoints of the chip, but only one endpoint at a time. For the endpoint you have selected, you must first program its type, direction and maximum packet size. Then assign it by setting the endpoint number in the DMA Endpoint register. This will cause the endpoint buffer to internally redirect to the DMA controller bus.

When setting the Endpoint Index register, make sure it is not configured with an endpoint number that has been used for DMA transfer. When set, the endpoint buffer of the selected endpoint is directed to the internal CPU bus for the PIO access.

3. Starting DMA

Dynamically assign the DMA Transfer Counter register for each DMA transfer. The transfer will end once the transfer counter reaches zero. Bit DMA_XFER_OK in the DMA Interrupt Reason register will be asserted to indicate that the DMA transfer has successfully stopped.

If the transfer counter is larger than the burst counter, the DREQ signal will drop at the end of each burst transfer. DREQ will reassert at the beginning of each burst. For a 32-bit DMA transfer, the minimum burst length is 4 bytes. This means that the burst length is only one DMA cycle. Therefore, DREQ and DACK will be toggled by each DMA cycle. For a 16-bit DMA transfer, the minimum burst length is 2 bytes.

Setting bit GDMA Read or GDMA Write in the DMA Command register will start DMA transfer.

4. DMA stop and interrupt handling

A DMA transfer will either be successfully completed or be terminated. To identify the interrupt source, the status in the DcInterrupt register and the DMA Interrupt Reason register must be read during the interrupt service routine.

If bit DMA_XFER_OK in the DMA Interrupt Reason register is asserted, it means the transfer counter has reached zero and the DMA transfer was successfully stopped.

If bit INT_EOT in the DMA Interrupt Reason register is set, it indicates that a short or empty packet was received. That is, DMA transfer was terminated. Normally, for an OUT transfer, it means that that the remote host wishes to terminate the DMA transfer.

If bits DMA_XFER_OK and INT_EOT are simultaneously set, it means the transfer counter reached zero and the last packet of the transfer is a short packet. Therefore, the DMA transfer was successfully stopped.

Setting bit GDMA Stop in the DMA Command register will force the DMA to stop, and bit GDMA_STOP in the DMA Interrupt Reason register will be set to indicate this event.

Setting bit Reset DMA in the DMA Command register will force the DMA to stop and initialize the DMA core to its power-on state.

5. Pesudo code

```
void Init_ISP1761_DC(void)
{
   /*
      Initilise all the Non-DMA releated ISP1761 DC registers
      here
   */
```

```
   //DMA Related Registers are initialized as follows

   Init_DCMode_Reg(DMAClkOn);           //(0x020C)
   Init_DMA_Burst_Cnt(BurstCount);            //(0x0264)
Init_DMA_HW_Reg(Hw_Config);          //(0x023C)
   Init_DMA_Config_Reg(DMA_Config);        //(0x0238)
   Init_Int_Ena_Reg(Interrupts_2B_Enable);       //(0x0214)
   Init_DMA_Int_Ena_Reg(DMA_Interrupts_2B_Enable); //(0x0254)

   /*Can program width of Edge Trigger here if using Edge Trigger.*/

}

void ARM_ISP1761DC_DMA(void)
{
   Init_DMA_TransferCounter(ActualTransferSize);   //(0x0234)
   Program_System_DMA();       //Arms system's DMA Master
   Init_DMA_Cmd_Reg(Arm_DC_DMA);        //(0x0230)
}
```

6. References

- *Universal Serial Bus Specification Rev. 2.0*
- *ISP1761 Hi-Speed Universal Serial Bus On-The-Go controller* data sheet

7. 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.

8. Contents

Date of release: 7 September 2004

Published in The Netherlands