AN10029

Odd or even byte indicator in the ISP1161A1

Rev. 03 — 12 October 2009 Application note

Document information

Info	Content
Keywords	isp1161; isp1161a1; usb; universal serial bus
Abstract	This application note describes the use of the odd or even byte indicator in the ISP1161A1.

Revision history

Rev	Date	Description
03	20091012	Rebranded to the ST-Ericsson template.
02	20090223	Rebranded to the ST-NXP Wireless template.
01	20030501	First release

Contact information

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

For document related queries, please send an email to: wired.support@stericsson.com

1. Introduction

The odd or even byte indicator allows firmware to read the exact number of data bytes that are transferred across the Direct Memory Access (DMA) transfer counter from the ISP1161A1 Device Controller (DC) to the external DMA bus. The odd or even byte indicator is valid only for the OUT token data.

The implementation will track whether an odd number has been transferred from the last OUT token packet to the DMA bus.

2. Location of the odd or even byte indicator

The odd or even byte indicator is implemented in bit 8 (ODD_EVEN_IND) of the DcDMAConfiguration register. The bit is read-only; see Table 1.

Table 1. DcDMAConfiguration register: bit allocation

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Symbol	CNTREN	SHORTP	reserved					ODD_EVEN_IND
Reset	0[1]	0[1]	0[1]	0[1]	0[1]	0[1]	0[1]	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	EPDIX[3:0]				DMAEN	reserved	BURSTL[1:0]	
Reset	0[1]	0[1]	0[1]	0[1]	0	0	0[1]	0[1]
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[1] Unchanged by a bus reset.

3. Behavior of the ODD_EVEN_IND bit

The ODD_EVEN_IND bit is logic 0 when the last DMA access is a byte (LSB Byte valid, MSB Byte invalid.). This bit is logic 1 when the last DMA access is a word (LSB Byte valid and MSB Byte valid.). You can treat this bit as **MSB Byte valid** for the last DMA access of 16 bits.

Table 2. Payload size for the ODD_EVEN_IND bit

ODD_EVEN_IND bit	**Payload size**
1	Even bytes
0	Odd bytes

4. Using the odd or even byte indicator for DMA

The odd or even byte indicator must be used together with the DcDMACounter register during SP_EOT to determine the correct amount of data transferred during a DMA read (OUT token).

The following formula can be used to calculate the number of bytes transferred by DMA.

Actual DMA transfer count = Internal DMA count – Bit 8 of the DcDMAConfiguration register

Number of bytes transferred by DMA = Programmed transfer count - Actual DMA transfer count

where Internal DMA count is the value in the DcDMACounter register at the end of DMA (SP_EOT or EOT)

Bit 8 of the DcDMAConfiguration register is the ODD_EVEN_IND bit

Programmed transfer count is the ‘Actual DMA transfer count’ programmed by the programmer before starting DMA

The result (Number of bytes transferred by DMA) will give the exact amount of data transferred by DMA (up to byte accuracy). You will be able to differentiate between whether the MSB Byte of the last word transferred by DMA is valid or not. If the result is odd, the MSB Byte is invalid; and if it is even, the MSB Byte is valid.

Note: The odd or even byte indicator is valid only for OUT endpoints.

4.1 Examples on the use of the odd or even byte indicator

4.1.1 Example 1

The host wishes to send **8 bytes** of data. The DC DMA counter is set to 8 bytes.

Actual DMA transfer = 1 – 1 (Even)

Number of bytes transferred by DMA = 8 – Actual DMA transfer

Number of bytes transferred by DMA = 8

4.1.2 Example 2

The host wishes to send **7 bytes** of data. The DC DMA counter is set to 8 bytes.

Actual DMA transfer = 1 – 0 (Odd)

Number of bytes transferred by DMA = 8 – Actual DMA transfer

Number of bytes transferred by DMA = 7

4.1.3 Example 3

The host wishes to send **34 bytes** of data. The DC DMA counter is set to 128 bytes.

Actual DMA transfer = 95 – 1 (Even)

Number of bytes transferred by DMA = 128 – Actual DMA transfer

Number of bytes transferred by DMA = 34

4.1.4 Example 4

The host wishes to send **33 bytes** of data. The DC DMA counter is set to 128 bytes.

Actual DMA transfer = 95 – 0 (Odd)

Number of bytes transferred by DMA = 128 – Actual DMA transfer

Number of bytes transferred by DMA = 33

5. Reading the DMA counter between transfers

The odd or even byte indicator is designed to give the correct value at the SP_EOT condition. If you, however, decide to read the DMA counter in between DMA transfers, see Table 3.

Table 3. Reading DMA counter between transfers

Device controller DMA counter	USB packet	Formula[1]
Even counter value	Even	Use formula
Even counter value	Odd	Use formula + 1
Odd counter value	Odd	Use formula + 1
Odd counter value	Even	Use formula

[1] Formula: Actual transfer = Programmed DC DMA counter – (read DC DMA counter – odd or even byte indicator)

5.1 Even counter examples

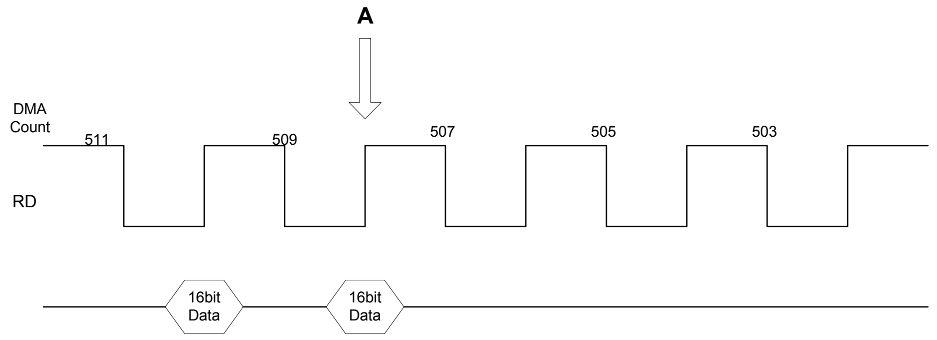

Fig 1. Even counter

5.1.1 Example 1

Consider that:

- DC DMA counter is 512 bytes
- USB transfers 10 bytes (OUT token)
- Odd or even byte indicator is 1(even byte).

If the micro wishes to read the ISP1161A1 DC DMA counter at A, then according to Table 3:

Actual transfer: 512 – (509 – 1) = 4 bytes

5.1.2 Example 2

Consider that:

- DC DMA counter is 512 bytes
- USB transfers 9 bytes (OUT token)
- Odd or even byte indicator is 0 (odd byte).

If the micro wishes to read the ISP1161A1 DC DMA counter at A, then according to Table 3:

Actual transfer: 512 – (509 – 0) + 1 = 4 bytes

5.2 Odd counter examples

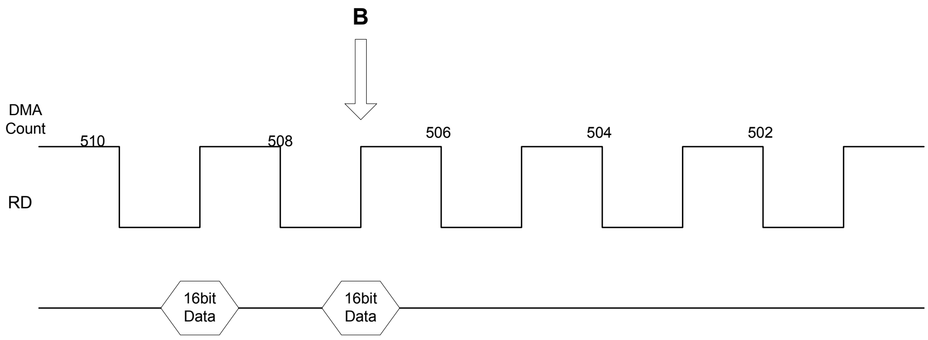

Fig 2. Odd counter

5.2.1 Example 1

Consider that:

- DC DMA counter is 511 bytes
- USB transfers 9 bytes (OUT token)
- Odd or even byte indicator is 0 (odd byte)

If the micro wishes to read the ISP1161A1 DC DMA counter at B, then as per Table 3:

Actual Transfer: 511 – (508 – 0) + 1 = 4 bytes

5.2.2 Example 2

Consider that:

- DC DMA counter is 511 bytes
- USB transfers 10 bytes (OUT token)
- Odd or even byte indicator is 1 (even byte).

If the micro wishes to read the ISP1161A1 DCDMA counter at B, then according to Table 3:

Actual Transfer: 511 – (508 – 1) = 4 bytes

6. Sending a zero-length packet

In the case of a zero-length packet, the Device Controller will generate an EOT interrupt.

Note: The interrupt is not SP_EOT.

6.1 Behavior of DREQ

If the ISP1161A1 device controller receives a zero-length packet, DMA will be terminated and the EOT interrupt will be asserted.

For example, consider that the device controller DMA counter is 512 bytes. The ISP1161A1 device controller will receive three packets of 64 bytes each, followed by a zero-length packet for termination. On receiving the zero-length packet, DREQ will not be asserted because DREQ assertion is based on whether the OUT buffer has data or not. Since it is a zero packet, it implies that it does not have data.

6.2 Behavior of the DMA counter

Consider a situation similar to that given in Section 6.1. If the ISP1161A1 device controller receives three full packets followed by an empty packet, and then an EOT Interrupt, it will result in correct DMA counter reading.

For example, consider that the DC DMA counter is 512 bytes. If the ISP1161A1 device controller receives three packets of 64 bytes each, followed by a zero-length packet for termination, reading the device controller DMA Counter will provide a value of 320 bytes (512 – 192 = 320). Therefore, the SP_EOT formula (Section 5) need not be applied.

6.3 Behavior of the ODD_EVEN_IND bit

On receiving the zero-length packet, the ODD_EVEN_IND bit will reflect as even.

7. References

[1] Universal Serial Bus Specification Rev. 2.0

[2] ISP1161A1 Full-speed USB single-chip host and device controller datasheet0

8. Legal information

Please Read Carefully:

The contents of this document are subject to change without prior notice. ST-Ericsson makes no representation or warranty of any nature whatsoever (neither expressed nor implied) with respect to the matters addressed in this document, including but not limited to warranties of merchantability or fitness for a particular purpose, interpretability or interoperability or, against infringement of third party intellectual property rights, and in no event shall ST-Ericsson be liable to any party for any direct, indirect, incidental and or consequential damages and or loss whatsoever (including but not limited to monetary losses or loss of data), that might arise from the use of this document or the information in it.

ST-Ericsson and the ST-Ericsson logo are trademarks of the ST-Ericsson group of companies or used under a license from STMicroelectronics NV or Telefonaktiebolaget LM Ericsson.

All other names are the property of their respective owners.

Contact information at www.stericsson.com under Contacts

www.stericsson.com

9. Contents

1. Introduction 3
2. Location of the odd or even byte indicator 3
3. Behavior of the ODD_EVEN_IND bit 3
4. Using the odd or even byte indicator for DMA .4
4.1 Examples on the use of the odd or even byte indicator..4
4.1.1 Example 1 4
4.1.2 Example 2 4
4.1.3 Example 3 4
4.1.4 Example 4 5
5. Reading the DMA counter between transfers ...5
5.1 Even counter examples......................................5
5.1.1 Example 1 6
5.1.2 Example 2 6
5.2 Odd counter examples 6
5.2.1 Example 1 6
5.2.2 Example 2 7
6. Sending a zero-length packet............................7
6.1 Behavior of DREQ...7
6.2 Behavior of the DMA counter 7
6.3 Behavior of the ODD_EVEN_IND bit 7
7. References 7
8. Legal information 8
9. Contents..9

For more information, please visit: http://www.stericsson.com
For document related queries, email to: wired.support@stericsson.com

Date of release: 12 October 2009
Document identifier: AN10029_3