

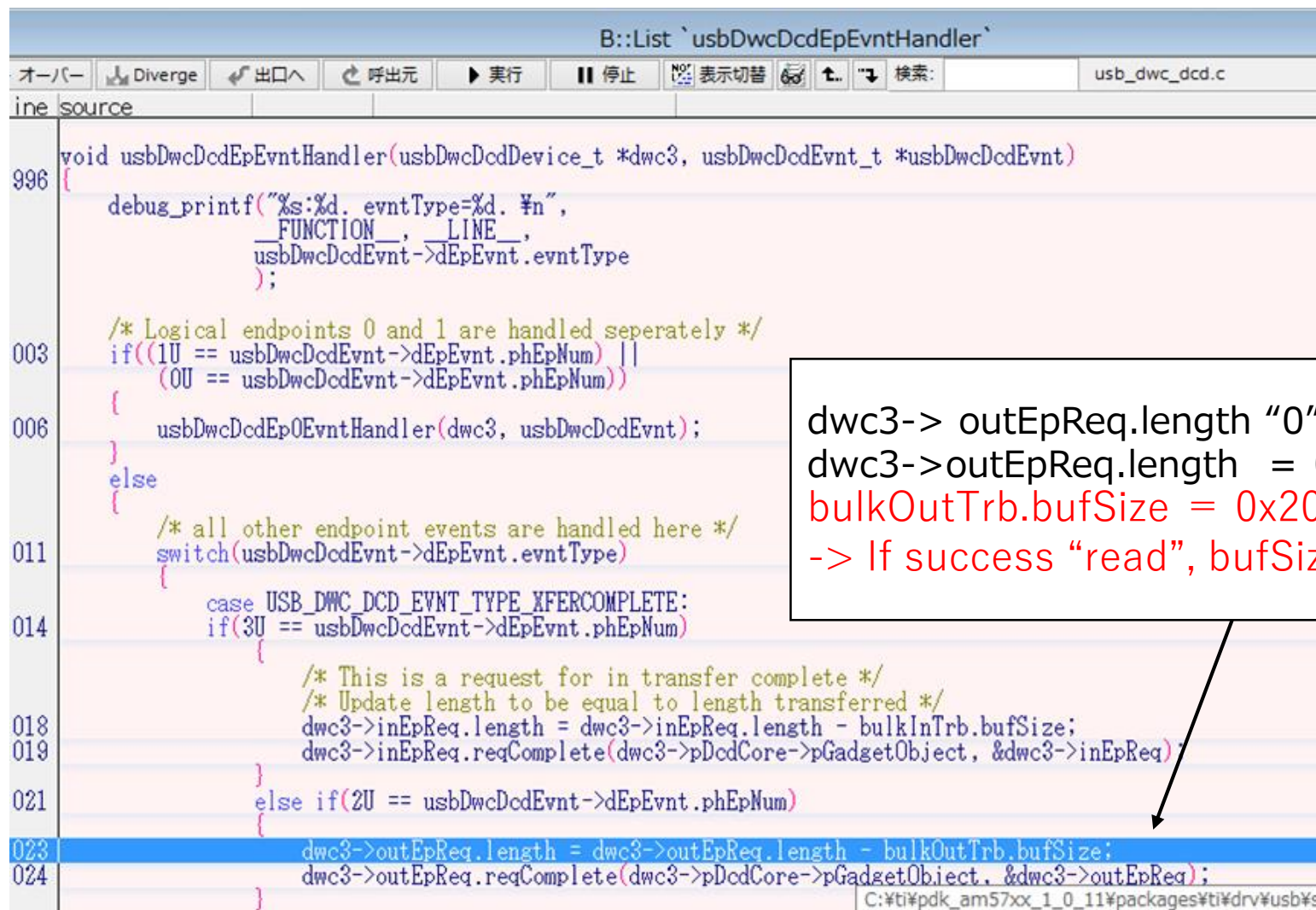
Issue: USB Read/Write sometimes fail

SOF=387(2/8)				4 439 676 900ns			
HS	I N	24	1	DATA1 00 7B FE 00 11 00 00 00 (37 bytes)	ACK	4 439 677 366ns	[.{]
SOF=387(3/8)-38E(4/8)				4 439 801 900ns			
HS	OUT	24	2	DATA0 5A 11 00 FF 8B FE FE 11 (20 bytes)	ACK	4 446 938 816ns	[Z.....]
				00 00 00 00 00 03 01 00			[.....]
				+10h 01 00 00 53			[...S]
SOF=38E(5/8)-38F(7/8)				447 052 233ns			
HS	I N	24	1	DATA0 5A 25 00 (3 bytes)	ACK	4 448 347 116ns	[Z%.]
SOF=38F(8/8)				4 448 427 300ns			
HS	I N	24	1	DATA1 00 8B FE FE 11 00 00 00 (37 bytes)	ACK	4 448 456 416ns	[.....]
SOF=390(1/8)-396(8/8)				4 448 552 316ns			
HS	OUT	24	2	DATA1 5A 11 00 FF 9B FE FE 11 (20 bytes)	ACK	4 455 529 616ns	[Z.....]
				00 00 00 00 00 03 01 00			[.....]
				+10h 01 00 00 43			[...C]
SOF=397(1/8)-397(2/8)				455 552 633ns			
<< EXTERNAL TRIGGER >>				4 455 678 050ns			

Information of protocol analyzer

Cause of failure

C:\ti\pdk_am57xx_1_0_11\packages\ti\drv\usb\src\dwc\usb_dwc_dcd.c



```
void usbDwcDcdEpEvtHandler(usbDwcDcdDevice_t *dwc3, usbDwcDcdEvt_t *usbDwcDcdEvt)
{
    debug_printf("%s:%d. evntType=%d. %n",
        __FUNCTION__, __LINE__,
        usbDwcDcdEvt->dEpEvt.evntType
    );

    /* Logical endpoints 0 and 1 are handled seperately */
    if((1U == usbDwcDcdEvt->dEpEvt.phEpNum) ||
        (0U == usbDwcDcdEvt->dEpEvt.phEpNum))
    {
        usbDwcDcdEp0EvtHandler(dwc3, usbDwcDcdEvt);
    }
    else
    {
        /* all other endpoint events are handled here */
        switch(usbDwcDcdEvt->dEpEvt.evntType)
        {
            case USB_DWC_DCD_EVT_TYPE_XFERCOMPLETE:
                if(3U == usbDwcDcdEvt->dEpEvt.phEpNum)
                {
                    /* This is a request for in transfer complete */
                    /* Update length to be equal to length transferred */
                    dwc3->inEpReq.length = dwc3->inEpReq.length - bulkInTrb.bufSize;
                    dwc3->inEpReq.reqComplete(dwc3->pDcdCore->pGadgetObject, &dwc3->inEpReq);
                }
                else if(2U == usbDwcDcdEvt->dEpEvt.phEpNum)
                {
                    dwc3->outEpReq.length = dwc3->outEpReq.length - bulkOutTrb.bufSize;
                    dwc3->outEpReq.reqComplete(dwc3->pDcdCore->pGadgetObject, &dwc3->outEpReq);
                }
            }
        }
    }
}
```

dwc3-> outEpReq.length "0"
dwc3->outEpReq.length = 0x200
bulkOutTrb.bufSize = 0x200 (Not updated)
-> If success "read", bufSize is updated

“bulkOutTrb” and “Receive buffer”

bulkOutTrb = (	
bufPtrLow = 0x85A9BE80,	
bufPtrHigh = 0x0,	
bufSize = 0x0200,	
packetCntM1 = 0x0,	
rsvd1 = 0x0,	
trbSts = 0x0,	
hwo = 0x1,	
lst = 0x1,	
chn = 0x0,	
csp = 0x0,	
trbCtrl = 0x1,	
ispImi = 0x0,	
ioc = 0x0,	
rsvd2 = 0x0,	
streamId = 0x0,	
rsvd3 = 0x0)	

Memory dump(Receive buffer)

address	0	1	2	3	4	5	6	7	01234567
NSD:85A9BE80	5A	11	00	FF	9B	FE	FE	11	ZNF8FFC
NSD:85A9BE88	00	00	00	00	00	03	01	00	ZUFBEET
NSD:85A9BE90	01	00	00	43	FF	FF	FF	FF	NNNNNESN
NSD:85A9BE98	FF	FF	FF	FF	FF	FF	FF	FF	UUUUUXHU
NSD:85A9BEA0	FF	FF	FF	FF	FF	FF	FF	FF	SNNCFFFF

*FF: Buffer filled before receiving

USB protocol analyzer

SOF=390(1/8)-396(8/8)	OUT	24	2	DATA1	5A 11 00 FF 9B FE FE 11 00 00 00 00 03 01 00 01 00 00 43	(20 bytes)
+10h						
<< EXTERNAL TRIGGER >>						

Data that could not be received was in the buffer.

TRB area is overwritten (Write back) with cache data. (Probably)

Therefore, the TRB area was set outside the cache area.

```

usb_dwc_dcd.c
/** ¥brief usb endpoint command parameters */
static usbDwcDcdDEpCmdParms_t dEpCmdParm __attribute__ ((aligned (ALIGNED_SIZE), section
(".bss:extMemNonCache:usbXhci")));

/** ¥brief TRB for transfer of control data */
static usbDEpTrb_t ctrlDataTrb __attribute__ ((aligned (ALIGNED_SIZE), section (".bss:extMemNonCache:usbXhci")));

/** ¥brief TRB for control status commands */
static usbDEpTrb_t ctrlStatusTrb __attribute__ ((aligned (ALIGNED_SIZE), section (".bss:extMemNonCache:usbXhci")));

/** ¥brief TRB for bulk in transfers */
static usbDEpTrb_t bulkInTrb __attribute__ ((aligned (ALIGNED_SIZE), section (".bss:extMemNonCache:usbXhci")));

/** ¥brief TRB for bulk out transfers */
static usbDEpTrb_t bulkOutTrb __attribute__ ((aligned (ALIGNED_SIZE), section (".bss:extMemNonCache:usbXhci")));

/** ¥brief USB event buffer */
static usbDwcDcdEvnt_t usbEvnt[USB_DCD_DWC_EVNT_BUF_MAX] __attribute__ ((aligned (ALIGNED_SIZE), section
(".bss:extMemNonCache:usbXhci")));

/** ¥brief Control data buffer */
static uint32_t usbCtrlDataBuf[32U] __attribute__ ((aligned (ALIGNED_SIZE), section (".bss:extMemNonCache:usbXhci")));

```

Added section (".bss: extMemNonCache: usbXhci") to __attribute__ of each memory definition.

* SUCCESS ALL
(READ 1000000 times, WRITE 2000000 times)

From this hypothesis and experiment,
we thought the TRB area was wrong.