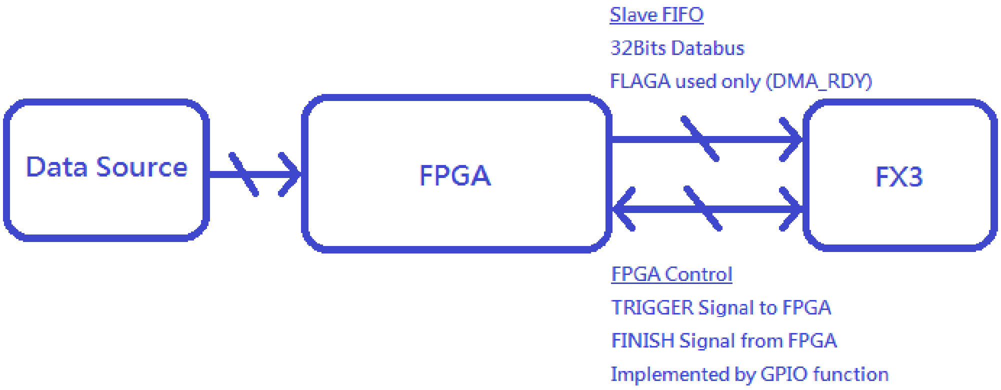
Dear Sirs,

Our design is to integrate FX3 with data source module. The data source module is bridged with a FPGA. The block diagram is as the following:



The clock to slave FIFO bus is 100MHz and setSysClk400=CyTrue is correctly configured by CyU3PDeviceInit() function. We use FLAGA flag which is dedicated to thread 0 (A1:A0 = “00”, use thread 0 only) to indicate DMA ready state for FPGA and we implement a counter mechanism in FPGA to satisfy DMA’s need. The data transfer process works well and got no any bus overrun/underrun errors.

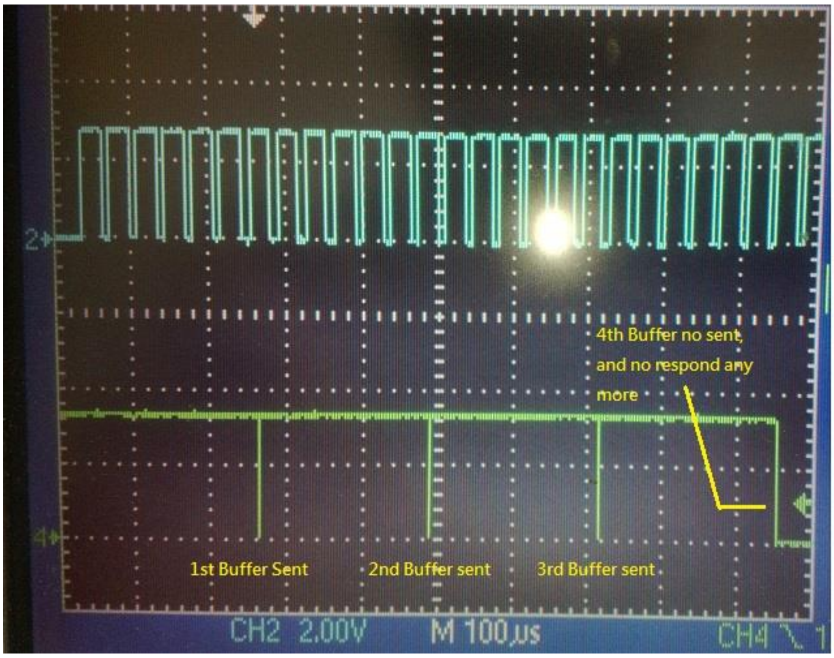
We adopt two endpoints one is interrupt EP1IN and the other is bulk endpoint EP2IN. The DMA channel is configured as :

```
/* Create a DMA MANUAL IN channel for P2U interrupt transfer. */
dmaCfg.size   = size;
dmaCfg.count = 2;
dmaCfg.prodSckId = CY_FX_PPORT_PRODUCER_SOCKET1;
dmaCfg.consSckId = CY_FX_USB_CONSUMER_SOCKET1;
dmaCfg.dmaMode = CY_U3P_DMA_MODE_BYTE;
dmaCfg.notification = CY_U3P_DMA_CB_PROD_EVENT;
dmaCfg.cb = IntrXferPtoUDmaCallback;
dmaCfg.prodHeader = 0;
dmaCfg.prodFooter = 0;
dmaCfg.consHeader = 0;
dmaCfg.prodAvailCount = 0;
apiRetStatus = CyU3PDmaChannelCreate (&g_hIntrPtoUChannel,
        CY_U3P_DMA_TYPE_MANUAL_OUT, &dmaCfg);
if (apiRetStatus != CY_U3P_SUCCESS)
{
    CyU3PDebugPrint (4, "CyU3PDmaChannelCreate failed, Error code = %d\n", apiRetStatus);
    CyFxAppErrorHandler(apiRetStatus);
}
```

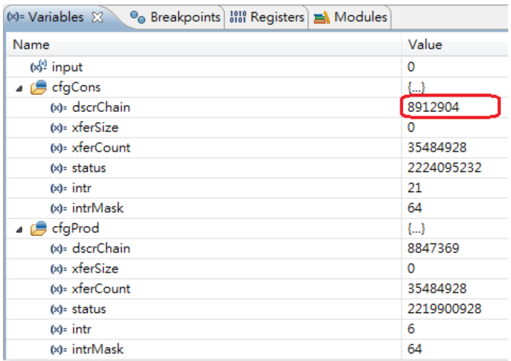
```
/* Create a DMA AUTO IN channel for P2U bulk transfer. */
dmaCfg.size    = size*16;
dmaCfg.count = 4;
dmaCfg.prodSckId = CY_FX_PPORT_PRODUCER_SOCKET2;
dmaCfg.consSckId = CY_FX_USB_CONSUMER_SOCKET2;
dmaCfg.dmaMode = CY_U3P_DMA_MODE_BYTE;
dmaCfg.notification = 0;
dmaCfg.cb = 0;
dmaCfg.prodHeader = 0;
dmaCfg.prodFooter = 0;
dmaCfg.consHeader = 0;
dmaCfg.prodAvailCount = 0;
apiRetStatus = CyU3PDmaChannelCreate (&g_hBulkPtoUChannel,
            CY_U3P_DMA_TYPE_AUTO, &dmaCfg);
if (apiRetStatus != CY_U3P_SUCCESS)
{
    CyU3PDebugPrint (4, "CyU3PDmaChannelCreate failed, Error code = %d\n", apiRetStatus);
    CyFxAppErrorHandler(apiRetStatus);
}
/* Flush the Endpoint memory */
CyU3PUsbFlushEp(CY_FX_EP1IN_CONSUMER);
CyU3PUsbFlushEp(CY_FX_EP2IN_CONSUMER);

/* Set DMA channel transfer size. */
apiRetStatus = CyU3PDmaChannelSetXfer (&g_hIntrPtoUChannel, 0);
if (apiRetStatus != CY_U3P_SUCCESS)
{
    CyU3PDebugPrint (4, "CyU3PDmaChannelSetXfer Failed, Error code = %d\n", apiRetStatus);
    CyFxAppErrorHandler(apiRetStatus);
}
apiRetStatus = CyU3PDmaChannelSetXfer (&g_hBulkPtoUChannel, 0);
if (apiRetStatus != CY_U3P_SUCCESS)
{
    CyU3PDebugPrint (4, "CyU3PDmaChannelSetXfer Failed, Error code = %d\n", apiRetStatus);
    CyFxAppErrorHandler(apiRetStatus);
}
```

The FPGA working scenario is : FX3 sends a TRIGGER signal to FPGA and then polls for FINISH signal activation from FPGA. Both TRIGGER/FINISH signals are implemented by FX3 GPIO. FPGA bursts in certain amount of data to EP2IN and interacts with FX3 DMA then assert FINISH when transfer is completed. We so far use EP2IN only and EP1IN not used.

The problem we are facing is : If we assert TRIGGER by a control transfer from PC (By invoking GPIO function in vender request events), the data transfer seems to be well. If we implement a CyU3PTimer with 1.0sec interval, and invoking GPIO function to assert TRIGGER in timer callback handler, the data transfer got fail. The transfer stopped in the last buffer of the buffer chain (even the first buffer chain is not completed) and no response any more. See the following picture:



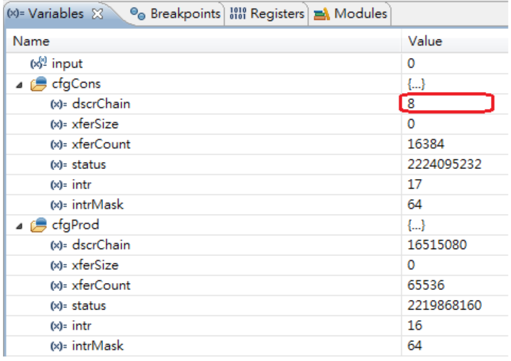
After we examine DMA channel / endpoints status, we got no error message and no error events. Further, we examine producer/consumer socket status by invoking CyU3PDmaSocketGetConfig after the DMA no responding, we got the following picture:



Name	Value
input	0
cfgCons	{...}
dscrChain	8912904
xferSize	0
xferCount	35484928
status	2224095232
intr	21
intrMask	64
cfgProd	{...}
dscrChain	8847369
xferSize	0
xferCount	35484928
status	2219900928
intr	6
intrMask	64

= 0x00880008

Consumer and Producer Sockets Configuration by Calling CyU3PDmaSocketGetConfig when transfer is successfully processed



Name	Value
input	0
cfgCons	{...}
dscrChain	8
xferSize	0
xferCount	16384
status	2224095232
intr	17
intrMask	64
cfgProd	{...}
dscrChain	16515080
xferSize	0
xferCount	65536
status	2219868160
intr	16
intrMask	64

= 0x00000008

Consumer and Producer Sockets Configuration by Calling CyU3PDmaSocketGetConfig when transfer is **NOT** successfully processed

As seen in the picture, we found that the consumer socket had no descriptors left to process but the successful case the consumer socket had lots of descriptors left. Any suggestions for us? What are the possible reasons for this phenomenon?

Thank you for your help!