

Hello

1. I need the data which length is 512 byte is sent to FX2LP over GPIF by the FPGA.
2. I am sure the data is being sent continuously by the FPGA. The ChipScope waveform shown in Figure 1 and Figure 2.

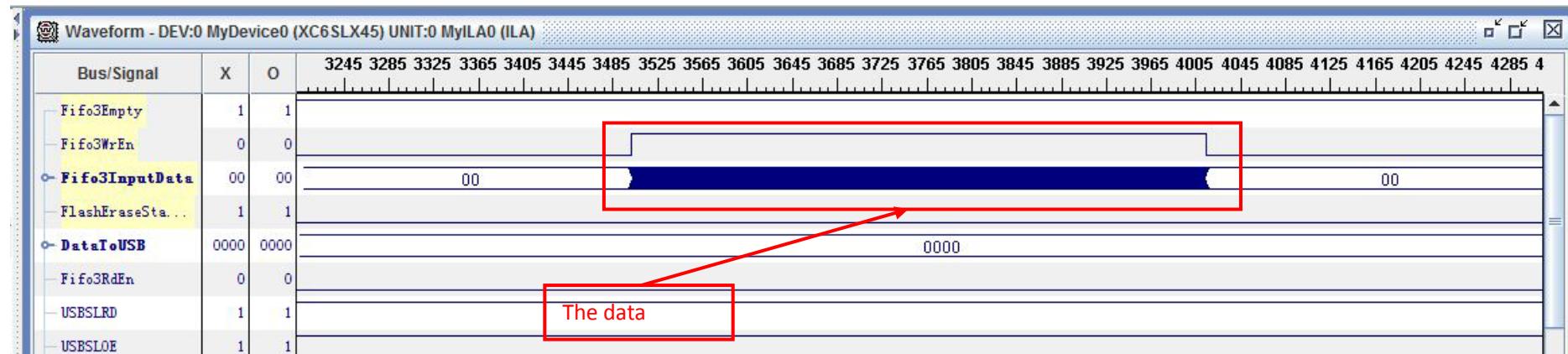


Figure 1

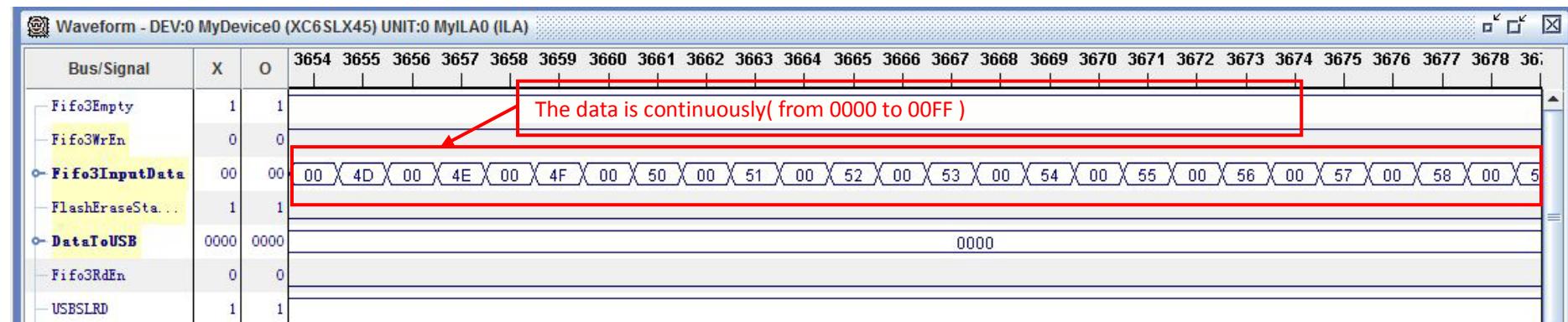


Figure 2

3. I have tried to use GPIF Byte Count as a flag, Figure 3, Figure 4, Figure 5 and Figure 6.

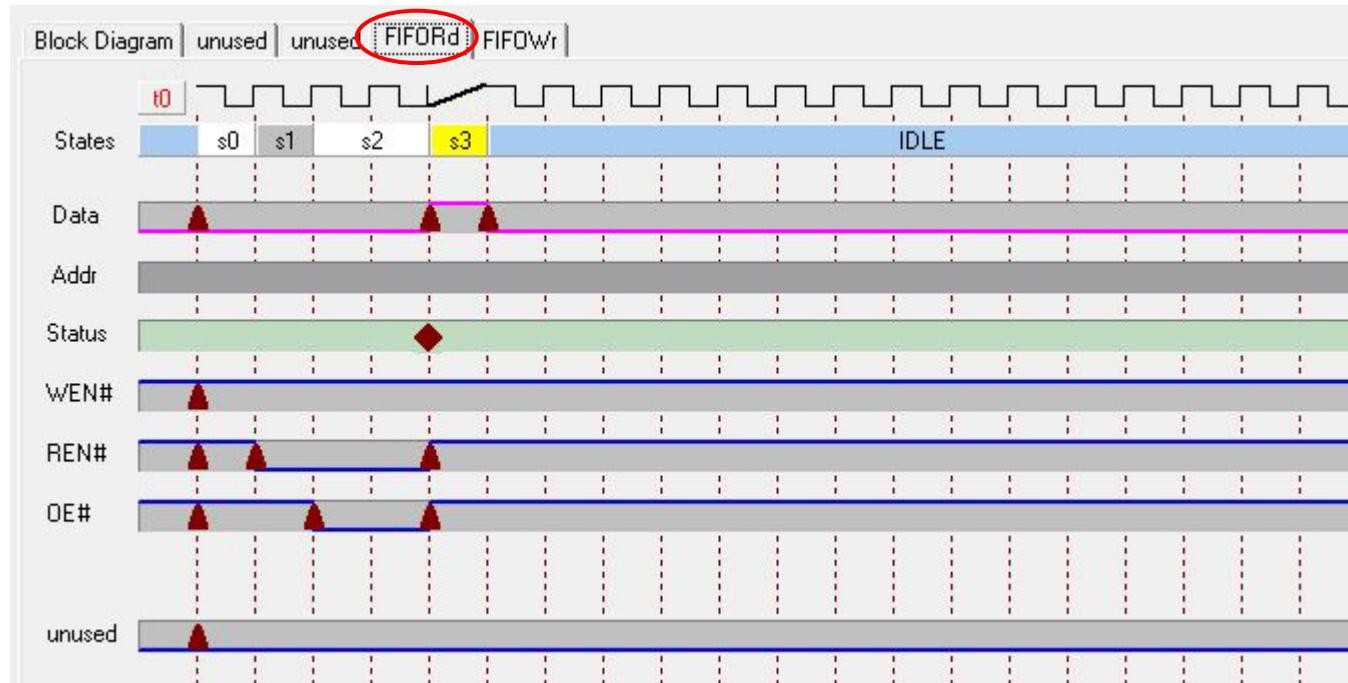


Figure 3

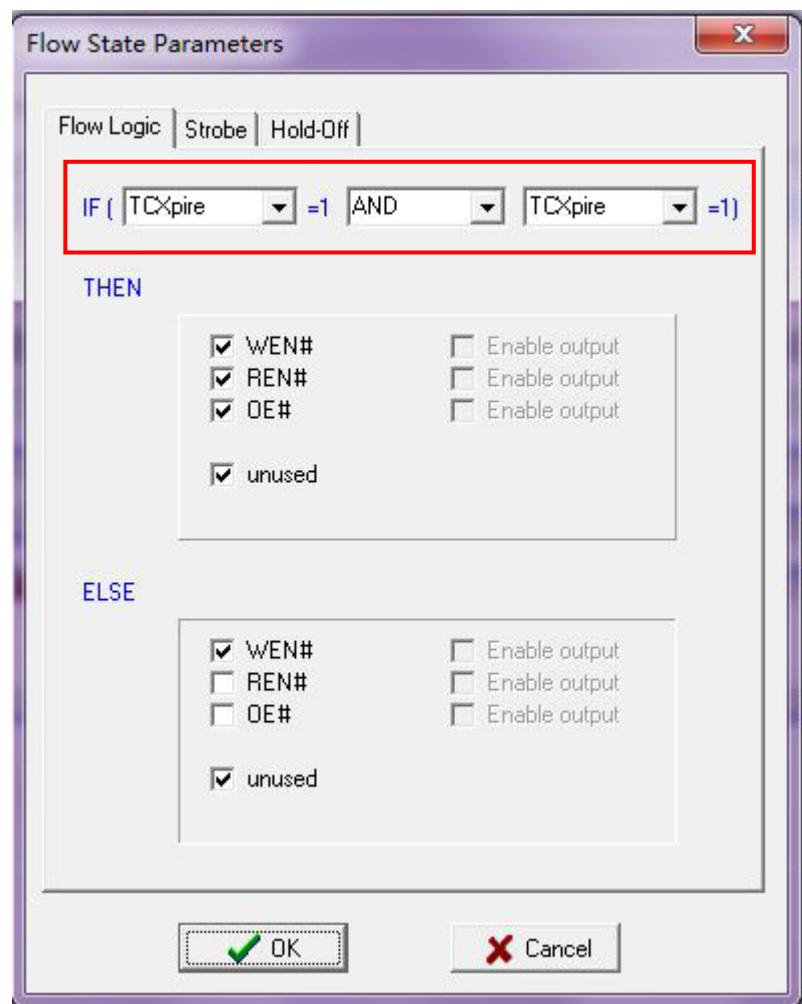


Figure 4

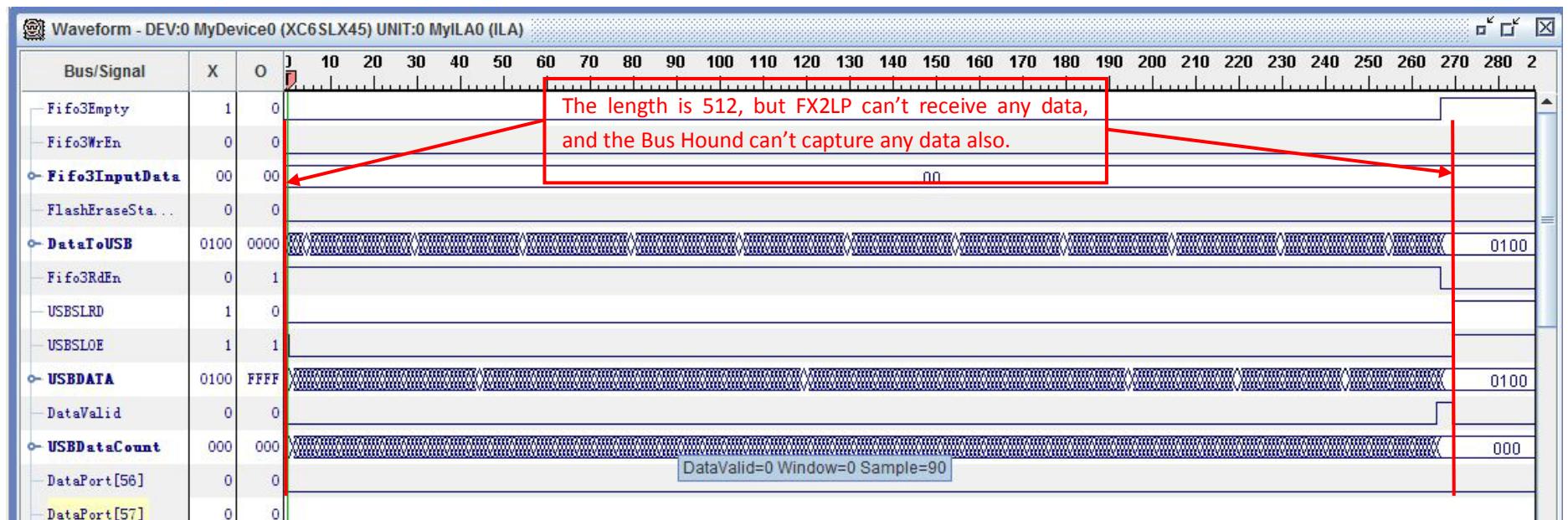


Figure 5

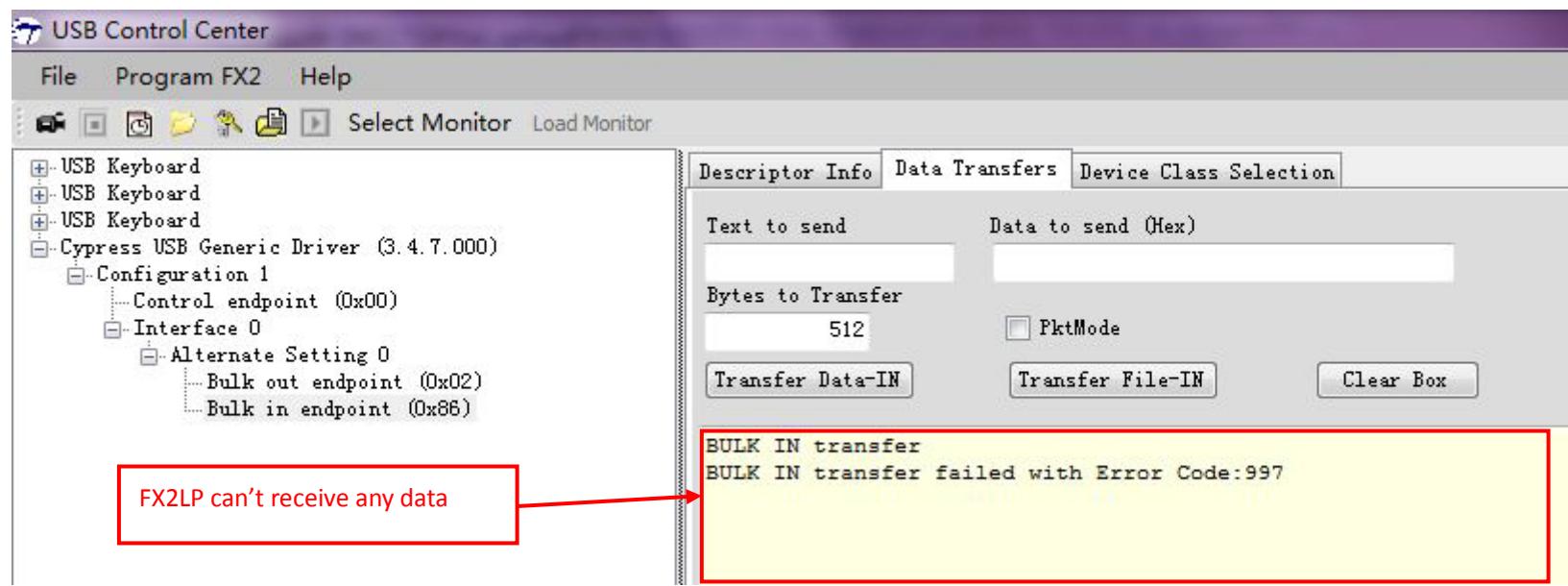


Figure 6

4. Figure 7 is the block diagram

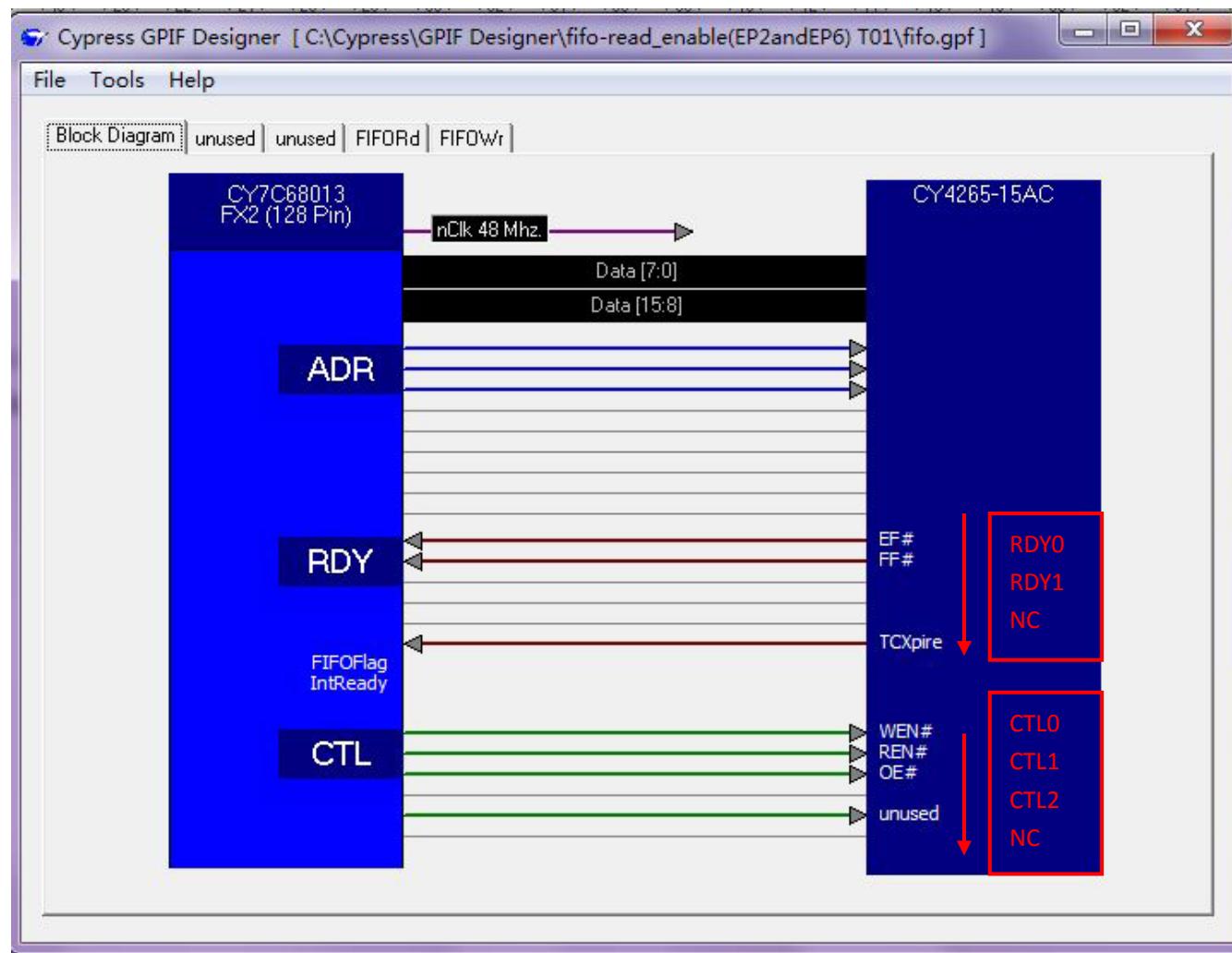


Figure 7

5. Figure 8 is the Status Action Points

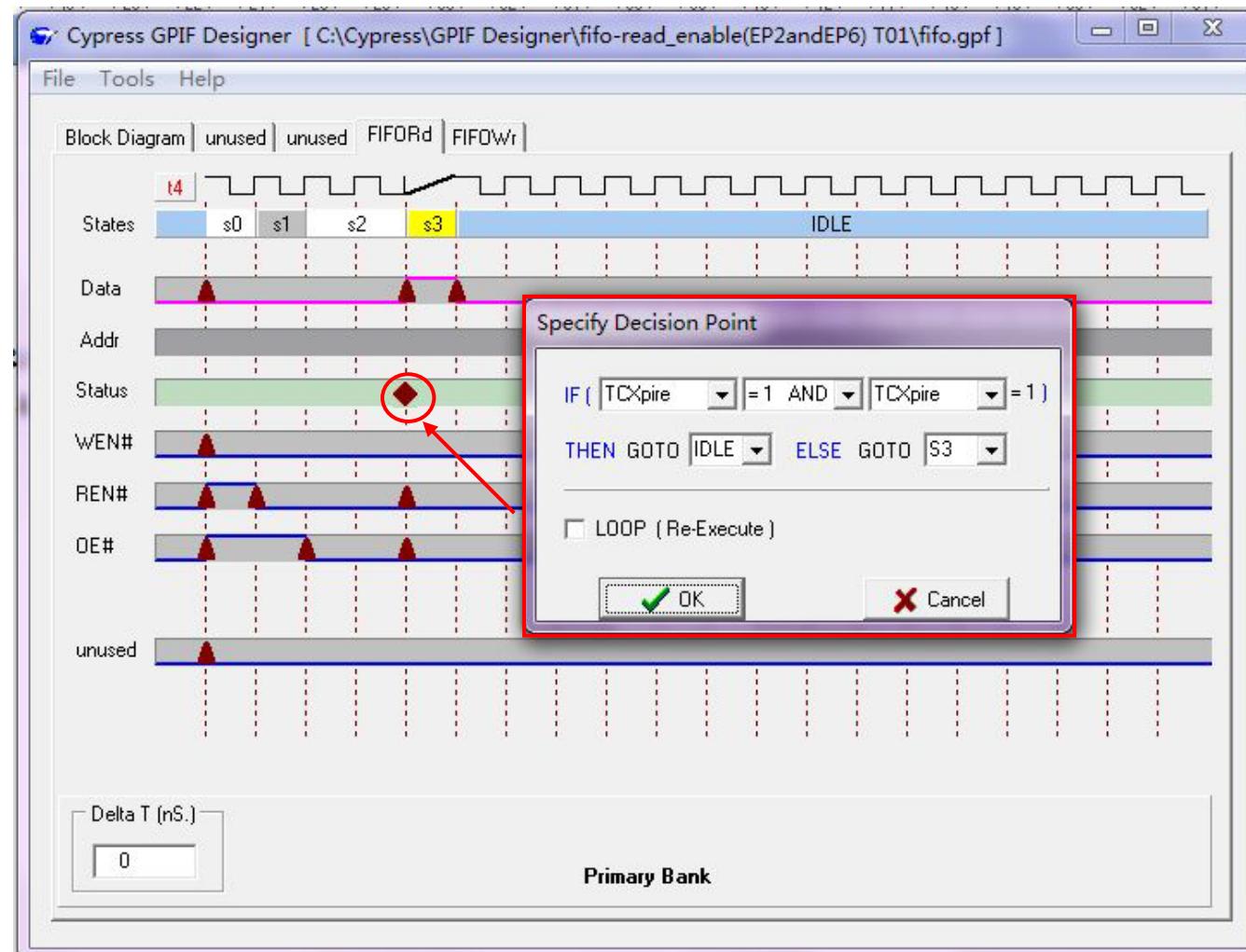


Figure 8

6. Part of the Program used for send and receive data

```
#define EXTFIFONOTFULL    GPIFREADYSTAT & bmBIT1
#define EXTFIFONOTEMPTY    GPIFREADYSTAT & bmBIT0

EP2CFG = 0xA0;      // EP2 is DIR = OUT, TYPE = BULK, SIZE = 512, BUF = 4x
SYNCDELAY;
EP4CFG = 0x00;      // EP4 is not valid
SYNCDELAY;
EP6CFG = 0xE2;      // EP6 is DIR = IN, TYPE = BULK, SIZE = 512, BUF = 2x
SYNCDELAY;
EP8CFG = 0x00;      // EP8 is not valid
SYNCDELAY;

FIFORESET = 0x80;   // set NAKALL bit to NAK all transfers from host
SYNCDELAY;
FIFORESET = 0x02;   // reset EP2 FIFO
SYNCDELAY;
FIFORESET = 0x06;   // reset EP6 FIFO
SYNCDELAY;
FIFORESET = 0x00;   // clear NAKALL bit to resume normal operation
SYNCDELAY;

EP2FIFO CFG = 0x01; // allow core to see zero to one transition of auto out bit
SYNCDELAY;
EP2FIFO CFG = 0x11; // auto out mode, disable PKTEND zero length send, word ops
```

```

SYNCDELAY;
EP6FIFO CFG = 0x01; // allow core to see zero to one transition of auto out bit
SYNCDELAY;
EP6FIFO CFG = 0x09; // auto in mode, disable PKTEND zero length send, word ops
SYNCDELAY;

void TD_Poll(void)
{
    if( GPIFTRIG & 0x80 )                                // if GPIF interface IDLE
    {
        if ( EXTFIFONOTFULL )                            // if external FIFO is not full
        {
            if ( !( EP24FIFOFLGS & 0x02 ) )           // if there's a packet in the peripheral domain for EP2
            {
                SYNCDELAY;
                GPIFTCB1 = 0x01;                         // setup transaction count (512 bytes/2 for word wide -> 0x0100)
                SYNCDELAY;
                GPIFTCB0 = 0x00;
                SYNCDELAY;

                Setup_FLOWSTATE_Write();                  // setup FLOWSTATE registers for FIFO Write operation
                SYNCDELAY;
                GPIFTRIG = GPIF_EP2;                     // launch GPIF FIFO WRITE Transaction from EP2 FIFO
                SYNCDELAY;
                while( !( GPIFTRIG & 0x80 ) )           // poll GPIFTRIG.7 GPIF Done bit
                {

```

```

        ;
    }

    SYNCDELAY;
} //End if (!(EP24FIFOFLGS & 0x02))
} //End if(EXTFIFONOTFULL)
} //End if( GPIFTRIG & 0x80 )

if(read_enable)
{
    if( GPIFTRIG & 0x80 )
    {
        if ( EXTFIFONOTEMPTY )           // if external FIFO is not empty
        {
            if ( !( EP68FIFOFLGS & 0x01 ) )      // if EP6 FIFO is not full
            {
                //      read_enable = FALSE;
                SYNCDELAY;
                EP6AUTOINLENH = 0x02;          // set AUTOIN packet length to 512 bytes
                SYNCDELAY;
                EP6AUTOINLENL = 0x00;
                SYNCDELAY;

                SYNCDELAY;
                GPIFTCB1 = 0x01;             // setup transaction count (512 bytes/2 for word wide -> 0x02)
                SYNCDELAY;
                GPIFTCB0 = 0x00;
}

```

```
SYNCDELAY;

Setup_FLOWSTATE_Read();           // setup FLOWSTATE registers for FIFO Read operation
SYNCDELAY;
GPIFTRIG = GPIFTRIGRD | GPIF_EP6; // launch GPIF FIFO READ Transaction to EP6 FIFO
SYNCDELAY;

while( !( GPIFTRIG & 0x80 ) )      // poll GPIFTRIG.7 GPIF Done bit
{
    ;
}

SYNCDELAY;
}//End if(!(EP68FIFOFLGS & 0x01))
}//End if(EXTFIFONOTEMPTY)
}//End if( GPIFTRIG & 0x80 )
}//End if(read_enable)
}//End void TD_Poll(void)
```