**NIOS AVALON USB2 based on FT2232**

**Abstract**

NIOS Avalon USB2 based on FTDI chip FT2232H Synchronous mode and theoretically can reach more than 25 Mbytes/s data transfer. Core contains 512 bytes read and write FIFO that can be modified for personal requirements.

**Added Features**

* IRQ based ON FIFO depths.
* IRQ generation on specific input symbol for receive only (NIOS side).
* Asynchronous FIFO reset (Receive FIFO or transmit FIFO)
* -- ??

**Registers**

Write data register address is **0x0**:

Read data register address is **0x1**.

TX FIFO LEFT status address is **0x2** it shows value of free TX FIFO space.

RX FIFO used status address is **0x3** it shows value of used RX FIFO.

Read status single bit of TX FIFO address is **0x4**.  
 (readdata(0)<=not( **tx\_wrfull**); this value is read separated for faster software run, it is inverted for simpler usage in C using if () { } command, useful for faster transmit)

Read status single bit of RX FIFO empty address is **0x5**.  
 (readdata(0)<=not(**rx\_wrempty**); this value is read separated for faster software run, it is inverted for simpler usage in C using if () { } command, useful for faster receive. If RX FIFO is not empty **rx\_wrempty=0** then read it)

Read **Global Status** register (Address **0x6**)it contains such values:

GLOBAL STATUS (0): -- WRITE FIFO FULL (if TX FIFO FULL then GLOBAL STATUS (0)->1 )  
GLOBAL STATUS (1): -- WRITE FIFO EMPTY ( if TX FIFO FULL then GLOBAL STATUS (1)->1 )  
GLOBAL STATUS (2): Reserved   
GLOBAL STATUS (3): Reserved  
GLOBAL STATUS (4): -- READ FIFO FULL  
GLOBAL STATUS (5): -- READ FIFO EMPTY  
GLOBAL STATUS (6): ENBALE\_IRQ\_RX\_NOT\_EMPTY  
GLOBAL STATUS (7): EVENT CHAR Interrupt (if irq occurs then ->1 if cleared in interrupt routine then ->0)

Set interrupt sensitivity value for RX FIFO. Address **0x7**.  
If you set this value to 32 then receive interrupt will be activated after it receives 32 bytes and etcetera.

Set END OF PACKED for RECEIVE FIFO address **0x8** enables interrupt on specific ASCII symbol. Related only for ASCII communication.

**Note: If you are attempting to clear interrupt in interrupt routine then you are resetting EVENT CHAR interrupt (IOWR\_8DIRECT(PORT\_0\_BASE+(9<<2), 0, 0);)**

**Clearing interrupt**

Interrupt is cleared at the end of interrupt routine writing 0x0 to address **0x9.**

**CONTROL REGISTER** (Write only) **Address 0xA**

0 bit – Writing 1 Enables IRQ\_COMPARE interrupt  
1 bit– Writing 1 Enables TX FIFO FULLL interrupt  
2 bit- Writing 1 Enables RX FIFO FULL interrupt  
3 bit.- Writing 1 Enables EVENT CHAR interrupt and sets GLOBAL\_STATUS(7)<='1';  
4 bit – Writing 1 Enables ENBALE\_IRQ\_RX\_NOT\_EMPTY interrupt  
5..7 bits reserved (writes are ignored)

**Asynchronous reset of FIFOS.**

In case that FIFOs are full and need to implement clear FIFO procedure, then using 0xB address and writing 0x1 transmit FIFO will be cleared. Writing 0x2 RX FIFO will be cleared.