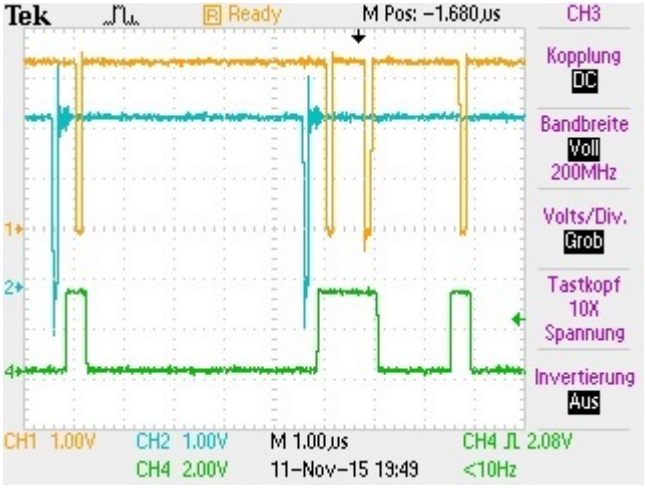
Hi,

I am working on a custom board with the STM32F746IGT6. in witch is a FTDI Parallel-USB converter is connected to the FMC.

My problem is that when I try to read data to "fast" from the FMC (after a while, and only sometimes) the FMC reads twice for one cycle. And the read function returns the result from the last transfer.
This error is only sometimes and I didn't find any relationship with the content of the data.

If I slow the velocity in them I pull data (> 2ms) then I have no more errors.

The following image shows the sporadically two reads at the same call.
The yellow line is the read trigger.
The blue line is the write trigger.
And the green line is an output pin that I trigger with software to recognize when the read function is called.



here is the code when I call a read:

```
01. static inline u8 FTDI_RX(void){
02.    return (*(__IO u8*) (NOR_MEMORY_ADRESS3 + FSMC_ADD_FTDI)); // (uint32_t)0x68000000 + 8
03. }
04.
05. char FTDI_READ(void){
06.    char tmp = 0;
07.    if (FTDI_canRead()) {          //READPIN(C, 14)
08.       SETPIN(H, 14);          //Green singal on
09.       tmp = FTDI_RX();
10.       RESETPIN(H, 14);        //Green singal off
11.       FTDI_mode = 1;
12.       return tmp;
13.    } else {
14.       return 0;
15.    }
16. }
```

The FMC configuration is based on the cube libraries:

```
01. NOR_HandleTypeDef hnor1;
02. FMC_NORSRAM_TimingTypeDef Timing;
03.   hnor1.Instance  = FMC_Bank1;
04.   hnor1.Extended  = FMC_NORSRAM_EXTENDED_DEVICE;
05.   //Duration of the address setup phase
06.   Timing.AddressSetupTime         = 7;    //0-15    4
07.   //Duration of the address hold phase
08.   Timing.AddressHoldTime          = 7;    //1-15    3
09.   //Duration of the data setup phase
10.   Timing.DataSetupTime              = 20;    //1-255    5
```

```
  //Duration of the bus turnaround phase
  Timing.BusTurnAroundDuration     = 5;    //0-15    1
  //Number of AHB clock cycles (HCLK) to build one memory clock cycle (CLK)
  Timing.CLKDivision             = 3;    //2-16    2
  //Number of clock cycles to issue to the memory before the first data of the burst
  Timing.DataLatency             = 3;    //2-17    2
  Timing.AccessMode             = FMC_ACCESS_MODE_A;
  /*
  - Data/Address MUX = Disable
  - Memory Type = SRAM
  - Data Width = 16bit
  - Write Operation = Enable
  - Extended Mode = Enable
  - Asynchronous Wait = Disable
  */
hnor1.Init.NSBank                 = FMC_NORSRAM_BANK3;
hnor1.Init.DataAddressMux         = FMC_DATA_ADDRESS_MUX_DISABLE;
hnor1.Init.MemoryType             = FMC_MEMORY_TYPE_SRAM;
hnor1.Init.MemoryDataWidth         = FMC_NORSRAM_MEM_BUS_WIDTH_16;
hnor1.Init.BurstAccessMode         = FMC_BURST_ACCESS_MODE_DISABLE;
hnor1.Init.AsynchronousWait     = FMC_ASYNCHRONOUS_WAIT_DISABLE;
hnor1.Init.WaitSignalPolarity     = FMC_WAIT_SIGNAL_POLARITY_LOW;
hnor1.Init.WaitSignalActive     = FMC_WAIT_TIMING_BEFORE_WS;
hnor1.Init.WriteOperation         = FMC_WRITE_OPERATION_ENABLE;
hnor1.Init.WaitSignal             = FMC_WAIT_SIGNAL_DISABLE;
hnor1.Init.ExtendedMode         = FMC_EXTENDED_MODE_DISABLE;
hnor1.Init.WriteBurst             = FMC_WRITE_BURST_DISABLE;
hnor1.Init.ContinuousClock         = FMC_CONTINUOUS_CLOCK_SYNC_ONLY;
hnor1.Init.WriteFifo             = FMC_WRITE_FIFO_DISABLE;             //
hnor1.Init.PageSize             = FMC_PAGE_SIZE_NONE;
// Initialize NOR control Interface
FMC_NORSRAM_Init(hnor1.Instance, &(hnor1.Init));
// Initialize NOR timing Interface
FMC_NORSRAM_Timing_Init(hnor1.Instance, &Timing, hnor1.Init.NSBank);
// Initialize NOR extended mode timing Interface
//FMC_NORSRAM_Extended_Timing_Init(hnor1.Instance, NULL, hnor1.Init.NSBank,
hnor1.Init.ExtendedMode);
// Enable the NORSRAM device
(hnor1.Instance)->BTCR[(hnor1.Init.NSBank)] |= FMC_BCR1_MBKEN;
```

I have already turned the D and I cache off, and got same results.

In the errata there is only an error for synchronous memories in burst mode. Witch I am not using.

STM32F74xxx STM32F75xxx Errata sheet

"2.5 FMC peripheral limitation
2.5.1 Dummy read cycles inserted when reading synchronous memories

Description
When performing a burst read access to a synchronous memory, two dummy read accesses are performed at the end of the burst cycle whatever the type of AHB burst access. However, the extra data values which are read are not used by the FMC and there is no functional failure.
Workaround None. "

Does some one have an Idea what I am doing wrong?

I apologize for my rusty English.

Best regards.
Robert