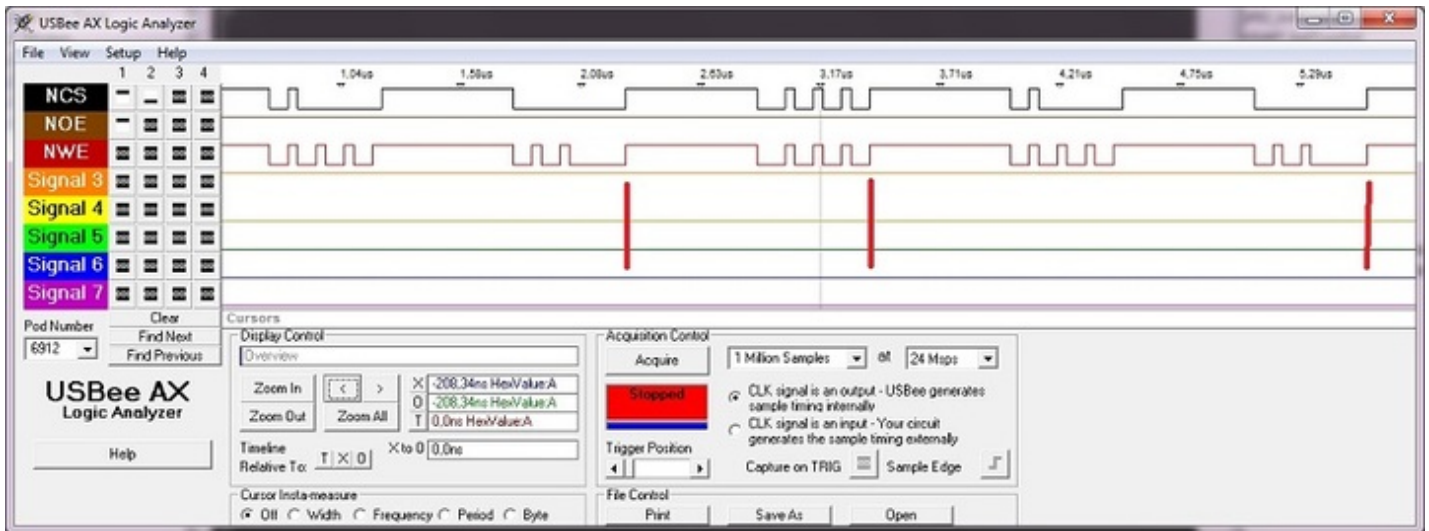


Hi. All!

I want to use FSMC module, but I'm not sure that it settings correct. There is write operation on a picture. I see multiple NWE activation. And sometime it release NEx together NWE (should be NWE and than NEx). What should I do for diagram like Figure 437 from [RM0090](#) ? Or, possible, I don't understand something?



Code:

```
typedef struct
{
    __IO uint8_t data;
} Modem_TypeDef;
#define MDATA ((uint32_t)(0x60000000))
#define MADDR ((uint32_t)(0x60000000 | 0x100))
#define M_DATA ((Modem_TypeDef *) MDATA)
#define M_ADDR ((Modem_TypeDef *) MADDR)

{
initFSMC();
    while(1)
    {
        GPIO_WriteBit( GPIOE, GPIO_Pin_0, Bit_SET); //my own strobe
        M_DATA->data = 0x77;
        M_ADDR->data = 0x33;
        GPIO_WriteBit( GPIOE, GPIO_Pin_0, Bit_RESET); //my own strobe
    }
}

void initFSMC( void)
{
    // --- initialization FSMC ---
    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOD | RCC_AHB1Periph_GPIOE, ENABLE); //clock for FSMC
    RCC_AHB3PeriphClockCmd( RCC_AHB3Periph_FSMC, ENABLE);

    // configure GPIOE [4:8]
    gpio.GPIO_Mode = GPIO_Mode_AF;
    gpio.GPIO_OType = GPIO_OType_PP;
    gpio.GPIO_Pin = GPIO_Pin_7 | GPIO_Pin_8 | GPIO_Pin_9 | GPIO_Pin_10 | GPIO_Pin_11;
    gpio.GPIO_PuPd = GPIO_PuPd_UP;
    gpio.GPIO_Speed = GPIO_Speed_25MHz;
    GPIO_Init( GPIOE, &gpio);

    // configure GPIOD [0,1,4,5,7,14,15]
    gpio.GPIO_Mode = GPIO_Mode_AF;
    gpio.GPIO_OType = GPIO_OType_PP;
}
```

```
gpio.GPIO_Pin = GPIO_Pin_0 | GPIO_Pin_1 | GPIO_Pin_4 | GPIO_Pin_5 | \
GPIO_Pin_7 | GPIO_Pin_14 | GPIO_Pin_15;
gpio.GPIO_PuPd = GPIO_PuPd_UP;
gpio.GPIO_Speed = GPIO_Speed_25MHz;
GPIO_Init( GPIOID, &gpio);

// timings
modemTimInit_str.FSMC_AddressSetupTime = 0; // used but don't care
modemTimInit_str.FSMC_AddressHoldTime = 0; // don't used
modemTimInit_str.FSMC_DataSetupTime = 5;
modemTimInit_str.FSMC_BusTurnAroundDuration = 0; // don't used
modemTimInit_str.FSMC_CLKDivision = 2;
modemTimInit_str.FSMC_DataLatency = 0; // don't used
modemTimInit_str.FSMC_AccessMode = FSMC_AccessMode_A;

FSMC_NORSRAMStructInit( &modemInit_str); // Apply default values
// Reset NOR/SRAM Initialization structure parameters values
modemInit_str.FSMC_Bank = FSMC_Bank1_NORSRAM1;
modemInit_str.FSMC_DataAddressMux = FSMC_DataAddressMux_Enable;
modemInit_str.FSMC_MemoryType = FSMC_MemoryType_PSRAM; //FSMC_MemoryType_SRAM;
modemInit_str.FSMC_MemoryDataWidth = FSMC_MemoryDataWidth_8b;
modemInit_str.FSMC_BurstAccessMode = FSMC_BurstAccessMode_Disable;
modemInit_str.FSMC_AsynchronousWait = FSMC_AsynchronousWait_Disable;
modemInit_str.FSMC_WaitSignalPolarity = FSMC_WaitSignalPolarity_Low;
modemInit_str.FSMC_WrapMode = FSMC_WrapMode_Disable;
modemInit_str.FSMC_WaitSignalActive = FSMC_WaitSignalActive_BeforeWaitState;
modemInit_str.FSMC_WriteOperation = FSMC_WriteOperation_Enable;
modemInit_str.FSMC_WaitSignal = FSMC_WaitSignal_Disable;
    modemInit_str.FSMC_ExtendedMode = FSMC_ExtendedMode_Enable; //FSMC_ExtendedMode_Disable;
modemInit_str.FSMC_WriteBurst = FSMC_WriteBurst_Disable;
modemInit_str.FSMC_ReadWriteTimingStruct = &modemTimInit_str;
modemInit_str.FSMC_WriteTimingStruct = &modemTimInit_str;

GPIO_PinAFConfig(GPIOIE, GPIO_PinSource7, GPIO_AF_FSMC);
GPIO_PinAFConfig(GPIOIE, GPIO_PinSource8, GPIO_AF_FSMC);
GPIO_PinAFConfig(GPIOIE, GPIO_PinSource9, GPIO_AF_FSMC);
GPIO_PinAFConfig(GPIOIE, GPIO_PinSource10, GPIO_AF_FSMC);
GPIO_PinAFConfig(GPIOIE, GPIO_PinSource11, GPIO_AF_FSMC);

GPIO_PinAFConfig(GPIOID, GPIO_PinSource0, GPIO_AF_FSMC);
GPIO_PinAFConfig(GPIOID, GPIO_PinSource1, GPIO_AF_FSMC);
GPIO_PinAFConfig(GPIOID, GPIO_PinSource4, GPIO_AF_FSMC);
GPIO_PinAFConfig(GPIOID, GPIO_PinSource5, GPIO_AF_FSMC);
GPIO_PinAFConfig(GPIOID, GPIO_PinSource7, GPIO_AF_FSMC);
GPIO_PinAFConfig(GPIOID, GPIO_PinSource14, GPIO_AF_FSMC);
GPIO_PinAFConfig(GPIOID, GPIO_PinSource15, GPIO_AF_FSMC);

FSMC_NORSRAMInit( &modemInit_str);
FSMC_NORSRAMCmd( FSMC_Bank1_NORSRAM1, ENABLE);
}
```