

NX1 使用 ICE 仿真的注意事项

- 内容:** NX1 使用 ICE 仿真时，当 NX1 停下时的周边动作说明。
- 原因:** 使用 NX1_FDB Ver.A/B 或 NX1_ICE Ver.A + NX1_COB VerA/B 的开发环境来侦错，当 ICE 停下来时，有些周边功能的动作并不会随着停止，以致于所取得的相关寄存器数值非当时停止时的数值。
- 方法:** 当使用这二个工具，建议用户将需要观察的数据透过 UART 界面上终端机进行观察动作，尽量不要使用 ICE Debug Mode 的停止功能。

[Example 1]

NX1_FDB Ver.A/B 开发环境，使用 ADC with Timer Trigger 功能，则当 NX1 停下时，此时若用户观察 ADC 寄存器的状态，则会发现 ADC 的 FIFO 已经满了及中断旗标会一直被举着，这是由于 Timer 依然持续动作，且触发 ADC 转换，造成短时间就填满了 FIFO。

上述条件，若再使用 Auto Scan Mode with FIFO 的条件下，则当 NX1 停下时，将可能使得 Channel 的控制顺序错乱。

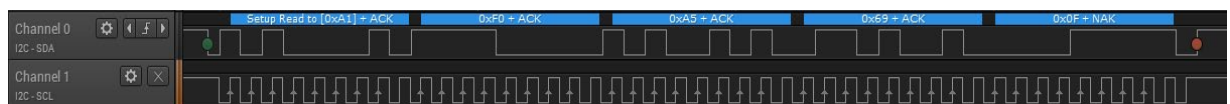
NX1_FDB Ver.C 和 NX1_ICE Ver.A + NX1_COB Ver.C 有针对 ICE Debug Mode 重新定义了周边功能的动作行为，让 NX1 周边也停下来然后再继续执行，以尽量和 Free Run 行为一致。

此版本的修改，用户需注意在应用于 I²C Slave Mode 时，当 NX1 停止后，此时 Master 发出 Read 指令，NX1 还是会持续执行 I²C 控制流程，不论 Master 要求多少笔信息，NX1 皆只会回复 FIFO 的第一笔数据，且因 FIFO Count 不会减少，后续再执行 Free Run 后，会因为 FIFO 的数据移位而造成错误。

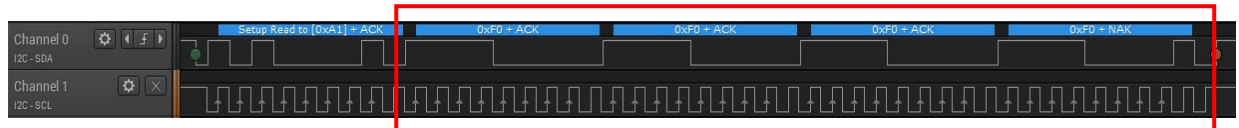
建议用户在 I²C 完成阶段传输后，再设置断点，若仿真必要于传送过程中设定断点时，用户在停下来看完状态后，建议让程序重新执行，以避免后续执行程序错误，此点必须注意。

[Example 2]

Free Run 实际状况: I²C FIFO 状态 Data[4] = { 0xF0 ,0xA5 ,0x69 ,0x0F }



ICE Debug Mode: 由于 EV chip 已停下来，此时 Master 来跟 NX1 (Slave)要数据，都只会得到 FIFO 的第一笔数据，如下图



Data 只传送目前 FIFO 当前第一笔数据，造成传送错误。