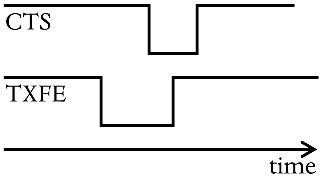
CS452 : Real-time Programming
Spring 2010
Train Controller Flow Control

Bill Cowan
University of Waterloo

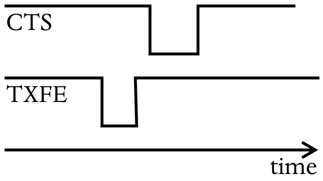
Students have had over the years a variety of problems which have the common symptom that occasional characters transmitted to the train controller get lost. This term, Spring 2010, Patrik Gornicz investigated this problem thoroughly, and has an explanation that can be used to eliminate the problem.

Two signals, CTS and TXFE in the Flag register, give the current state of the transmit logic: CTS is asserted when the train controller is ready to receive a byte; TXFE is asserted when the transmitter is in the process of transmitting a byte.[1] When the UART is enabled with the FIFO disabled, the two signals are asserted as shown in the figure to the right.



1. TXFE is negated when a byte is written to the data register of the UART.
2. TXFE re-asserted when transmission of the byte is complete.
3. CTS is negated by the train controller some time after the byte is written, possibly before, possibly after TXFE is re-asserted.
4. CTS is later re-asserted by the train controller, possibly before, possibly after the re-assertion of TXFE.

After a byte has started transmission to the train controller it is safe to write the next byte only after both CTS and TXFE have been re-asserted. Note that it is not possible just to check the CTS and TXFE bits: the timing might be like the



[1] When the FIFO is disabled, and the UART enabled, the BUSY bit should be the inverse of the TXFE bit.

case in the figure to the right, in which case a byte transmitted the first time that both bits are set would be lost.

Interrupts occur when TXFE is asserted and when CTS is negated or asserted. It is possible to write the next byte to the data register by polling frequently enough that you never miss the short time when CTS is negated. Thus, you should expect to get three interrupts from the transmitter of the UART between the time when you write a byte to the data register and when it is safe to write the succeeding byte. However, I do not recommend counting interrupts, but using the interrupts to drive a simple state machine associated with the UART.