

FPGA 初心者は自作マイコンの夢を見るか？

RISC-V on FPGA

実装計画

第2回 UARTの実装

中森 章

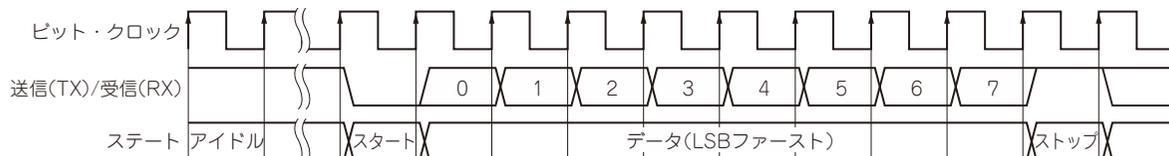


図1 UART送受信のタイミング。送信(TX)と受信(RX)で全く同じ

はじめに

前回(2022年6月号)は、FPGA(Field Programmable Gate Array)の中に、RISC-VのCPUコア(TwosomeCore)と汎用入出力のGPIO(General-purpose I/O)を実装することで、初めてのFPGA体験を解説しました。

今回からはさらに、自作マイコンの完成に向かって周辺ユニットを実装していきます。その第1弾はUART(Universal Asynchronous Receiver/Transmitter)です。

● UARTとは何か

UARTは、多目的非同期送受信という意味で、標準的な周辺機能の1つです。UARTを搭載していないマイコンは滅多にありません。自作マイコンにUARTを搭載するのは必然です。

UARTには送信用に1本、受信用に1本と合計2本の信号線しかありません。基本的にTX(送信信号)とRX(受信信号)の2線を使ってマイコンと外部機器との通信を行います。送受信のタイミングを調整するために、RTS(Request To Send)やCTS(Clear To Send)という信号線が使われることもあります。これらの信号は、GPIOで代用することもできます。送受信のタイミングをCPUが管理している場合は、特に必要な信号でもないので、今回の実装でもRX信号とTX信号の2本のみの対応にします。

● UARTの通信方法

UARTは多くの場合、8ビット長のデータをシリアルに転送します。「非同期」という名称が付いていますが、これは、シリアル転送するビットをクロックに

同期させずに行うことに由来します。

クロックに同期した転送方式では、クロック線が追加され、送信側はクロックの立ち上がり(または立ち下がり)のタイミングでデータを1ビットずつ送り出し、受信側はクロックの立ち上がり(または立ち下がり)のタイミングでデータを1ビットずつ受け取ることで通信を確立します。

UARTのように、クロック線が存在しない場合は、データの1ビットの転送にかかる時間を、送信側と受信側で一致させておくことで、データを正しく送受信できます。この1ビットにかかる時間をボー・レートという指標で決定します(正確には、ボー・レートは1秒間に転送するビット数のこと)。

● UARTの送受信のタイミング

図1にUARTの送受信のタイミングを示します。送信時も受信時も転送形式(プロトコル)は同一です。図1では、TXとRXのみが実在する信号線です。「ビット・クロック」や「ステート」は仮想的なものです。ビット・クロックの1周期が、上述のボー・レートになります。

TX信号もRX信号も転送がないとき(ステートがアイドルのとき)は“H”レベルになります。そして、転送を開始するときに、TX信号を1ビット期間の“L”レベルに駆動します。受信側は、RX信号が“H”レベルから“L”レベルに変化したときに転送が開始されたことを認識します。その後は、8ビット分のデータを転送(ビットの値が“1”のときは“H”レベルを駆動、“0”のときは“L”レベルを駆動)して、最後に1ビット期間“H”レベルを駆動します。この最後の1ビットはストップ・ビット(stop bit)といい、転送の終了を示します。図1では