

CPUがなくても
仮想シリアル・
ポートが
組み込める!

接続初期化処理(エnumレーション)をハードウェアが自動で行う USBコミュニケーション・デバイス・クラス対応の USBターゲット機器の実装

石丸 顕二 Kenji Ishimaru

USBシステムを実現するのに一番やこしいところは、接続初期化処理(エnumレーション)部分でしょうか。このエnumレーションをハードウェアが自動で勝手に行ってくると、ソフトウェアの作成は非常に楽になるでしょう。ここではUSBエnumレーション処理をロジック化した、仮想シリアル・ポートの実装事例を紹介します。前編の今回は、プロセッサを使わない、完全ロジック制御型USB-CDCを解説します。

1 ロジック制御とプロセッサ制御

● USBコミュニケーション・デバイス・クラスを実装

USB仮想シリアル・ポート(仮想COMポート)は、ホストと開発ボード間のデータ送受信などに用いられます。例えば、図1のように、ホストと開発ボードをシリアル・ポートで接続したシステムで、開発ボードの動作状況をターミナル・ソフトウェアで確認する場合などに用いられます。

このようなシステムの場合、ホストと開発ボード間でやりたいことは、モニタ・プログラムへのコマンドの送信や実行ログの取得といったデータ送受信です。しかし、データ送受信を行うためには、まずホストが行うエnumレーションと呼ばれる処理に対して、ターゲット機器がUSBコミュニケーション・デバイス・クラス(Communications Device Class, 以下USB-CDC)として応答する必要があります。

● OpenCoresのUSB IPコアを紹介

FPGA マガジンNo.8の記事⁽¹⁾では、ターゲット機器側に必要なエnumレーションに対する応答処理などをOpenCoresのUSB 1.1 Host and Function IP core(以降USB IPコア)と、ソフトコア・プロセッサの組み合わせで実現しました。

しかしFPGAリソースの問題で「プロセッサは使いたくない」、またはプロセッサを使う場合であっても「エnumレーション処理の負担を少なくしてプロセッサは別の処理に集中したい」といった場合もあるでしょう。

「エnumレーション処理の負担を少なくしてプロセッサは別の処理に集中したい」といった場合もあるでしょう。

● ロジック制御でエnumレーションを行う

そこでここでは、ロジック制御でエnumレーションを行う仮想シリアル・ポートの実装事例を紹介します。具体的には、仮想シリアル・ポートの機能をすべてロジック制御で行う事例と、エnumレーションのみロジック制御で行い、データ送受信はソフトコア・プロセッサで行う事例の2種類を紹介します。ここではこれらを、

- ・完全ロジック制御型(プロセッサなし)
- ・ロジック制御+プロセッサ制御型
(エnumレーションのみロジック処理)

と呼ぶことにします。

なお、FPGA マガジンNo.8で解説している事例を、ここでは便宜的に、

・完全プロセッサ制御型

(USB IPコアの制御は全てプロセッサが行う)

と呼ぶことにします。つまり、USB-CDCを実現する構成として、図2に示す3種類が考えられます。

● 3種類の構成

図2の一番上の「完全プロセッサ制御型」は、OpenCoresのUSB IPコア(usbSlave)とソフトコア・プロセッサで仮想シリアル・ポートの機能を実現した、FPGA マガジンNo.8で解説したシステム構成です。

図2の中段は「ロジック+プロセッサ制御型」です。RTLで記述したロジック制御部とUSB IPコアをまとめたusbCdcTargetというラッパ・モジュールを作成し、それを新たなIPコアとしてFPGA内でソフトコア・プロセッサと接続した構成です。ソフトコア・プロセッサはエnumレーションを意識する必要がなく、データ送受信やそのほかの処理に集中できます。

図2の一番下は「完全ロジック制御型」です。USB IPコアとロジック制御部をまとめたusbCdcDeviceという

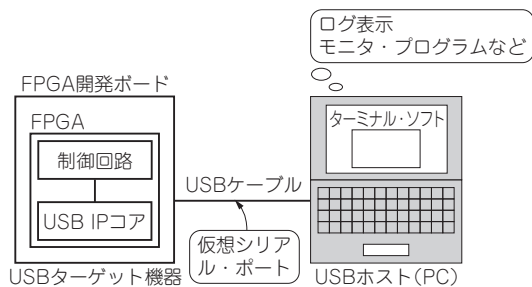


図1 仮想シリアル・ポートのシステム例