

準備：環境構築とLチカ

鈴木 量三朗

Tang Nano 9K は RISC-V を意識した低価格 FPGA ボード

Tang Nano 9K のプラスチック・ケースには、Entry-Level FPGA Kit For RISC-V と銘打たれており、RISC-V が動く FPGA というのがこのボードのウリのようです。

● RISC-V を載せてソフトウェアで制御もできる

一度 FPGA 上に RISC-V のような CPU を実装すれば、C 言語などを使ってソフトウェアを開発することで、Tang Nano 9K の制御ができます。

この場合はソフトウェア・プログラマにとっては RISC-V の知識は必要になるものの慣れ親しんだプログラミングによる開発ができるでしょう。

ただし、ソフトウェアだけでは FPGA のメリットを生かすような処理は記述しにくくなります。

● ハードウェアの開発には HDL を使う

FPGA ボード上に回路を作る場合には、HDL (Hardware Description Language) を使ってデザインします。

RISC-V のサンプル^{注1}では AXI Lite のバスを採用しており、拡張しようと思うと、バス上に回路を実装することになります。これは、ソフトウェア・プログラマにとっては敷居が高く感じるかもしれません。

エントリー・レベル (初心者向け) と考えた場合は RISC-V や AXI Lite を利用せず、HDL だけでも十分に Tang Nano 9K の開発ができます。

例えば、LED やスイッチの操作については、HDL の教科書を見ながら比較的素直に扱えるでしょう。

● 価格が手ごろなので試しやすい

このボードの使いどころの1つはやはり電子工作でしょう。値段も手頃でブレッドボードに挿して周辺回路などを拡張可能です。

将来的に下の基板 (ベース・ボード) を作るということも視野に入れつつ FPGA としてどう料理していくのかを考えると夢も膨らみます。

注1: このサンプルは <https://github.com/sipeed/TangNano-9K-example/tree/main/picotiny> にあります。

試しやすいけれど HDL で開発? …ちょっと敷居が高い

● 実開発では HDL の複雑なモジュール構築が欠かせない

一方でボードには多くの周辺機器が搭載されています。HDMI や SD カード・スロット、32M ビット SPI フラッシュ・メモリ、PSRAM (疑似 SRAM) などのデバイスを生かそうと思うと HDL による複雑なモジュール構築が欠かせません。

例えば、USB 経由で UART を使いたくてもマイコンのように簡単にはいきません。FPGA にシリアル通信用の特別なハードウェアが必要であり、その制御もしなければなりません。

● C 言語なら Hello World も楽なんだけど

FPGA の提供する柔軟さの1つがクロック・レベルでの時間的処理解像度の高さや並列性であるなら、CPU が提供するとは抽象的なプログラミング環境でしょう。100MHz で動く FPGA なら 10ns オーダで処理を記述できます。

対して CPU と C コンパイラがあれば、printf で “Hello World” を出力するのも容易でしょう。

この2つの柔軟性は二律背反、トレードオフの関係です。

高位合成で FPGA をソフトウェアっぽく使おう

Tang Nano 9K はせっかくの手頃な FPGA です。その長所と短所の中間をうまくバランスを取りながら開発をしたいものです。FPGA の開発では、高位合成と IP (Intellectual Property) コアがそのような柔軟性を提供します。

● Python で FPGA の回路を記述する

ここで紹介する Polyphony (ポリフォニー) は、Python で記述したコードをコンパイルすると、Verilog HDL 記述を生成するものです。

生成された Verilog HDL は、実際に FPGA で動く回路として合成可能です。Python のようなソフトウェア言語に基づくコードから、FPGA などで動作可能な

本書サポート・ページのご案内。ダウンロード・データも下から入手できます。
<https://interface.cqpub.co.jp/fpga02/>