誰でもプログラミング言語Cで回路が作れる

高位合成言語Impulse Cによる画像処理アルゴリズムの実装
ハードウェア/ソフトウェアの協調動作とパイプラインの最適化

宮島 敬明 Takaaki Miyajima

FPGAマガジンNo.1ではImpulse Cを中心に，高位合成の大まかな概要を述べました．今回は，Impulse Cのソフトウェア上でのシミュレーションを付属のサンプル・コードを元に実践的で高度な使い方を解説します．

1 FindNemoプログラム

● FindNemoプログラムの概要

Impulse Cには様々なサンプル・コードが付属しています．FindNemoは，Example/Imageフォルダに収録された，有名な3Dアニメーション映画の主人公であるクマノミを映像内から探し出し，スポット・ライトで照らすサンプル・プログラムです（**図1**）．このプログラムは，Nios IIのようなソフト・コアとFPGAのハードウェア・モジュールが協調動作するように作られ，**図2**に示すように大きく三つのモジュールから構成されています．

それぞれの役割は，consumerプロセスが入力データを用意，filterプロセスが実際のクマノミ探索，producerプロセスが出力画像を生成します．画像データ（実際はRGBの画素値）は，FIFOを介して次のプロセスへ送られます．このような設計モデルはProducer-Consumerモデルと呼ばれ，Impulse Cでは好んで利用されています．もともとは，マルチスレッド・プログラミングの同期制御の設計モデルですが，高位合成ではそれぞれのプロセスが並列動作させるハードウェアを模しているため，このモデルが適しているとされています．

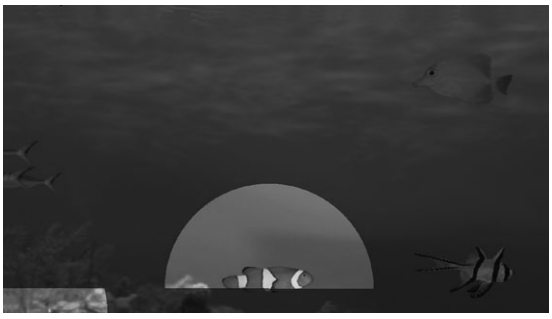

図1 FindNemoサンプル・プログラム

● ハードウェア/ソフトウェア協調動作

ハードウェア/ソフトウェア協調動作を実現するように，三つのモジュールはそれぞれ動作する場所が異なっています．ソフトウェア・プロセスであるconsumerプロセスとproducerプロセスはFPGA上のソフト・コア上で動作し，ハードウェア・プロセスであるfilterプロセスのみがFPGA上のハードウェア・モジュールとして動作します．ソフトウェア・プロセスはprintf関数など，C言語を完全にサポートしていますが，ハードウェア・プロセスは制限を持ったC言語で記述します．

Impulse Cではハードウェア・プロセスとソフトウェア・プロセスの指定は非常に簡単で，co_process_config関数と呼ばれる関数でハードウェア・プロセスにしたいものを選ぶだけです．本サンプル・プログラムでは，fish_filter_hw.cの275行目にあるconfig_impulse_DVI関数の内部で，

```
co_process_config(filter_process,
co_loc, "PE0");
```

と宣言されています．これで，filter_processがハードウェア・プロセスとなり，HDLが自動生成されます．

2 シミュレーション時の動作とFPGA上での動作の切り替え

● シミュレーション時とFPGA上の動作

Impulse Cでは，ハードウェア/ソフトウェア協調動作だけでなく，シミュレーション時の動作とFPGA上で異なった動作させることも可能です．FindNemoプログラムではfilterプロセスはそのままですが，

図2 FindNemoプログラムのプロセス図