Vol.2019-EMB-51 No.6
2019/6/21

高位合成可能なC++記述のための分散ミドルウェアを用いた機能シミュレーション

新井 健太[1,a)]　大川 猛[2]　大津 金光[1]　横田 隆史[1]

キーワード：FPGA，コンポーネント指向開発，publish/subscribe 通信

1. はじめに

組込みシステムにおいて画像処理等を高速かつ省電力で実現するデバイスとして FPGA(Field Programmable Gate Array) が注目されている．近年の FPGA 開発では RTL(Register Transfer Level) 設計だけでなく，ソフトウェアの言語 (C, C++など) の記述を HDL(Hardware Description Language) に合成する高位合成も取り入れられている [1]．いずれの手法においても，要求を満たすハードウェアを実現するためには，アルゴリズムに内在する並列性を引き出して実装する必要がある．一方で FPGA はチップによってハードウェアリソース量や動作周波数が異なるため，ソフトウェアに比べてモジュールの移植性や再利用性が乏しい．

そこで我々は FPGA のコンポーネント指向開発手法を提案した [2]．この手法は実装済みの回路コンポーネントを組み合わせてハードウェアを開発するため，回路の再利用性を向上させる．

この手法では各コンポーネントの単体テストを C++によるシミュレーションで行っていたが，一方で統合テストは RTL シミュレーションで行う必要があった．一般に HDL を使った開発はソフトウェアの言語を使った場合よりも開発時間が長期化する問題がある．

そこで本稿では，C++による複数コンポーネントの統合シミュレーション手法を提案する．シミュレーションに分散通信ミドルウェアである DDS(Data Distribution Service)[3] を使用することで，コンポーネント間でのデータ通信が可能となる．すなわち本手法は C++の記述のままでの統合シミュレーションを可能とする．本稿では Xilinx 社の高位合成ツール Vivado HLS 向けの C++記述への提案手法の適用について説明する．

1　宇都宮大学大学院工学研究科情報システム科学専攻
2　東海大学情報通信学部組込みソフトウェア工学科
a)　kenta@virgo.is.utsunomiya-u.ac.jp

2. publish/subscribe 通信フレームワーク

publish/subscribe 通信フレームワークは，FPGA の回路をコンポーネント指向で実装するためのフレームワークである [2]．フレームワークを図 1 に示す．このフレームワークに採用した publish/subscribe 通信モデルは，ノード間が疎結合なためにノードの追加や機能変更による他のノードへの影響が小さく，モジュールが再利用性しやすい利点がある．開発者は FPGA で実現する機能をノードとして実装し，それらをトピックを介して接続する．

トピックは publisher(送信側のノード) が送信したメッセージを全ての subscriber(受信側のノード) に配信する．この通信モデルでは，subscriber がメッセージを受け取れない状態であっても，publisher はメッセージを送信することができる．またメッセージは publisher が送信した順に配信される．

フレームワークを使って実装したノード間で publish/-subscribe 通信をするために，トピックに対応したモジュールが必要である．トピックモジュールを図 2 に示す．トピックは FIFO バッファとスイッチで構成される．バッファを設けることで，先述の非同期通信を実現する．またスイッチは入力側の FIFO を順番に確認し，メッセージを見つけ次第出力側の FIFO に書き込む．メッセージは出力側の全ての FIFO に書き込まれる．

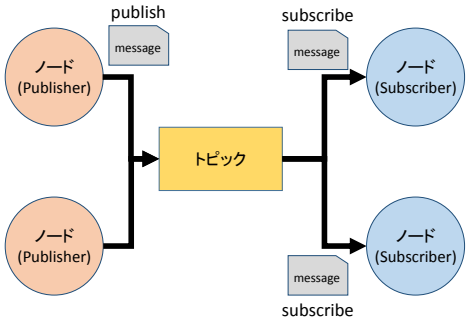

図 1　publish/subscribe 通信フレームワーク

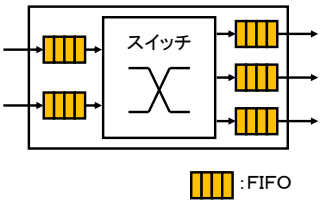

図 2　トピックモジュール (2 入力 3 出力)

3. DDS を用いた機能シミュレーション

DDS は publish/subscribe 型通信を行うための通信ミドルウェアの仕様である．DDS ミドルウェアは近年の分散システムにおけるデータ通信に利用されている．DDS ミドルウェアを使った通信には通信相手を指定する必要がなく，通信するプロセスの追加や削除が容易である．

publish/subscribe 通信フレームワークへ実装するノードは，トピックへの publish/subscribe をトピックモジュールの FIFO に対する読み書きで実現する．ノードを Vivado HLS で合成可能な C++記述として実装する場合，FIFO への読み書きに hls::stream 型を使用できる．hls::stream 型のオブジェクトはメンバ関数 read(), write() で FIFO に対し読み書きの操作を行う．これは C シミュレーションでは無限深さの FIFO として扱われ，合成時には FIFO とのインターフェースとして合成される．

C シミュレーションの際に，hls::steam 型を使った FIFO にアクセスで他のシミュレーションされているノードに対して publish/subscribe 通信することができれば，複数のノードを統合シミュレーションすることが可能だと考えた．そこで C シミュレーション中のノードが FIFO に読み書きする際に，代わりに DDS の publish/subscriber 通信を使用する．DDS を用いたシミュレーションの構造を図 3 に示す．各ノードのプロセスがトピックに対して publish もしくは subscribe する場合に DDS の機能が呼び出される．DDS が publisher の配信したメッセージを各 subscriber へ配信することで，FPGA に実装するモデルと同じようにノード間でメッセージがやり取りされる．

hls::stream 型をラップして publish/subscribe 通信の機能を追加したトピック型を実装し，簡単な publisher と subscriber を実装した．例として publisher のソースコードを図 4 に示す．publisher は subscriber に対し sample 型のメッセージをやりとりする．各ノードを Vivado HLS の C シミュレーションで実行したところ，publisher のメッセージを subscriber が受信できた．またこの C++記述を HDL に合成することができた．

4. おわりに

本稿では，C++による複数モジュールの統合シミュレーション手法を提案した．モジュール間の publish/subscribe 通信に DDS を使用することで，複数モジュールを統合した際の機能レベルでのシミュレーションを実現した．ハードウェア化する C++記述は，シミュレーションのための変更後もそのままハードウェア化が可能である．

今後の課題として，更なる開発効率の改善のために，テストしたコンポーネントを自動的に統合して FPGA 上に実装することが挙げられる．

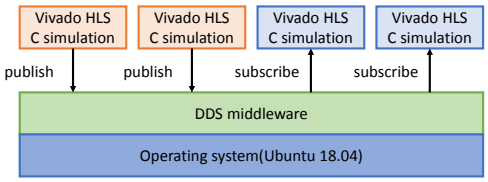

図 3　DDS を使ったシミュレーション

```
1  #include "sample_publisher.hpp"
2  namespace {
3      sample message;
4  }
5  void sample_publisher(
6      SampleTopic &st
7  ) {
8      message.data(message.data() + 1);
9      st.receive(message);
10 }
```

図 4　publisher の実装

参考文献

[1] Razvan Nane, Jongsok Choi, Yu Ting Chen, Fabrizio Ferrandi: A Survey and Evaluation of FPGA High-Level Synthesis Tools, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, VOL.35, NO.10, Oct. 2016.

[2] Kenta Arai, Takeshi Ohkawa, Kanemitsu Ootsu, Takashi Yokota: Component-based FPGA development of intelligent image processing for industrial IoT devices, 2nd International Workshop on Embedded Software for Industrial IoT, Mar. 2019.

[3] Object Management Group: Data Distribution Service(DDS), 入手先 ⟨https://www.omg.org/spec/DDS/1.4/PDF⟩, (2015.04.10).