ISS-P-020

ハードウェア記述言語 FSL における手続き記述手法の拡張

中村　一歩†　　渡邊　誠也†　　名古屋　彰†
† 岡山大学大学院自然科学研究科

1. はじめに

近年，ディジタルシステムでは高性能かつ低消費電力な処理が求められている．それを実現する手段の一つとして特定機能のハードウェア化があり，それに伴いハードウェア設計の重要性が高まっている．ハードウェア設計においてはハードウェア記述言語（Hardware Description Language，以降HDLと略す）が使用され，我々はFSL（Functional and Scalable hardware description Language）と名付けたHDLを開発中である [1].

現在，FSLでハードウェアの複数サイクルの処理を記述する場合，1マシンサイクルに閉じた処理を記述する関数と複数マシンサイクルの処理を記述するステージを組み合わせて記述する必要がある．本研究ではこれをより簡潔に記述できるように手続き記述の構文procを導入する．

2. ハードウェア記述言語 FSL

FSL は生産性の高いハードウェア設計環境の実現を目指したHDL で，Verilog HDL などの HDL と比べて抽象度の高い記述が可能である．FSL の特徴としてハードウェアを手続き的モデルで記述することが挙げられる．手続き的モデルはハードウェアの動作を手続きのやり取りで表現するもので，制御の流れが理解し易い記述が可能である．

3. 提案手法

図 1 に現在の複数サイクル処理の記述法と提案する記述法を示す．現在は def で定義した関数からステージを起動し，その中で複数サイクルの処理を記述しなければならない（図 1 (a)）．これを簡略化するため手続き記述の構文 proc を導入する（図 1 (b)）．この proc 構文により複数サイクルにわたる手続きを記述する．

図 2 は proc で定義した手続きを seq ブロックの中から呼び出す場合の動作イメージを示している．seq はブロック内の処理を 1 サイクル毎に逐次的に実行していく構文である．処理 A が終了した後，proc で定義された手続き p が呼び出される．手続き p 中の処理 B の終了を待たずに後続の処理 C が進められる．処理 B は処理が終了した際に r に結果を格納する．処理 C が終了した後，処理 B の結果を利用する処理 D が開始される．このときに，呼出し側は処理Bの結果がrに格納されているかを確認する．結果が格納されている場合は処理 D を待ち時間なく開始することができる．処理 B が終了しておらず，結果が r に格納されていない場合は，処理 B の終了を待つ．

def で定義した関数の呼出しでは，呼び出した同一サイクルで結果を受け取るが，proc で定義した手続きは同一サイクルで結果を受け取れるとは限らない．そのため proc の手続きを呼び出した際は結果を格納する入れ物を受け取る．この入れ物を後続の処理において参照する際に，結果が格納されているかどうかで後続の処理を進めるか，あるいはブロックするかを判断する．

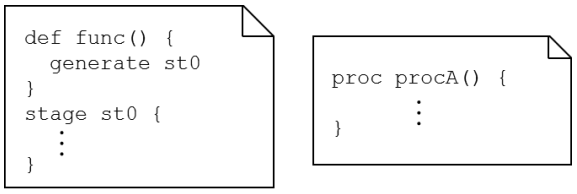

図 1．複数サイクル処理の記述

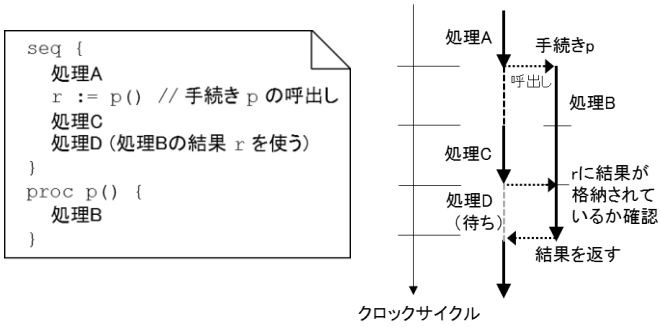

図 2. proc 構文で定義した手続きの動作イメージ

4. 合成するハードウェア

proc構文で定義した手続きを呼び出す記述から合成するハードウェアについて述べる．procで定義した手続きを呼び出す場合，処理結果を格納するレジスタと，手続きの終了のフラグを示すレジスタを呼び出し側に用意する．処理結果を参照する際は，このフラグを監視して処理の終了を判断する．処理が終了していれば結果をレジスタから取り出し処理を進め，終了していなければ終了を待つ．

5. 今後の課題

今後はここで述べた proc 構文の実装を進めていく．また proc 構文を使ったハードウェア実装を行い，記述の有用性を検証することも課題として挙げられる．

参考文献

[1] 渡邊 誠也，名古屋 彰，“オブジェクト指向/関数型言語をベースとするハードウェア記述言語 FSL の設計，”電子情報通信学会技術研究報告，RECONF2015-37，pp. 27－32，2015.