(様式 5)

2020 年 12 月 9 日
Year Month Day

学位（博士）論文要旨

(Doctoral thesis abstract)

論文提出者 (Ph.D. candidate)	工学府博士後期課程 電子情報工学専攻 Department of Electronic and Information Engineering (major) 2018 年度入学 (Admission year) 学籍番号 18834303 氏名 照屋 大地 (student ID No.) (Name) Daichi Teruya
主指導教員氏名 (Name of supervisor)	中條拓伯 Hironori Nakajo
論 文 題 目 (Title)	ディジタル回路自動設計に適したプログラミングモデルと言語の研究 Practical programming models and languages for digital circuit design automation

論文要旨 (2000 字程度)
(Abstract(400 words))
※欧文・和文どちらでもよい。但し、和文の場合は英訳を付すこと。
(in English or in Japanese)

ASIC や FPGA で用いるディジタル回路の設計は，ハードウェア記述言語 (hardware description language; HDL) を用いたレジスタ転送レベル (Resister Transfer Level; RTL) による開発が必要となる．記述の性質から HDL はバグが発生しやすく，デバッグにも様々なツールが必要で熟練した技術が必要である．そこで C/C++ や Java といったプログラミング言語によって記述されたプログラムやアルゴリズムと同じふるまいをする回路を合成する技術である高位合成 (high-level synthesis; HLS) と呼ばれる技術が注目され既に様々な言語による HLS ツールが開発されている．しかしながらソフトウェア開発と回路設計の記述モデルには大きな隔たりがある．ソフトウェアのプログラムには並列性やデータフローに関する情報が含まれておらず，最適化のための情報を付け加えなければスループットが非常に低くなってしまうことや，複雑で巨大なステートマシンが生成されてしまい，自動的な並列化やパイプライン化が難しい問題が発生する．

そこで本研究は，FPGA を主なターゲットとして，ソフトウェア開発の分野で生産性を高めるために用いられているプログラミングモデルによってディジタル回路設計の生産性を高めることを目的とした研究を行った．本研究は大きく分けて 1) フレームワークを用いた最適なアーキテクチャの自動生成ツールの研究，2) ディジタル回路設計に近いプログラミングパラダイムを用いた回路設計ツールの研究の二つのアプローチに分けられる．

まずフレームワークによる設計支援ツールに取り組んだ．センサ入力が必要となる組み込みシステムにおいて，一般的な汎用マイクロコントローラ (MCU) では通信速度やセンサ数が増えるにつれデータハンドリングが困難になる．そこで，ユーザが定義したセンサ入力やデータ処理の一部を FPGA へオフロードしたアーキテクチャを自動的に生成するフレー

ムワーク，PyJer を提案した．生成アーキテクチャを限定的にすることで既存の最適化手法やツールの自動的な適用が行いやすく，ユーザによる専門的知識に基づくコードの最適化を最小限に抑えることができる．評価のため簡易的な 2 次元ビームフォーミングによる音源方位推定システムを構築した.作成したシステムのコード量は,Java が 398 行,Python が 30 行，Verilog HDL が 118 行となった．実機検証の結果，ビームフォーミング処理の動作レートは約 174[Hz]と CPU で動作させた場合の 691[Hz]と比較して低速なものとなったが，センサとの通信は全て並列化することができるためセンサの数が増えたとしても処理レートは安定する．また， FPGA の LUT 使用量が 10%程度，BRAM の使用量が 15¥%程度と低くなった．CPU-FPGA 間通信のための機構などによるリソース消費のオーバヘッドを小さく抑えながらもセンサとの通信の並列化と実用上十分なデータ処理速度を達成することができた．

PyJer の中では Java 言語ベースの HLS ツールを利用している．HLS ツールを用いてソフトウェアとして利用されてきた既存のプログラムから回路を合成しようとした場合には，性能が低下してしまうことが一般的である．性能低下を回避するためには回路設計の知識と HLS ツールにおける技法を熟知したエンジニアによるコードの最適化が必要となる．筆者は，ほとんどの HLS ツールにおいてプログラムの最適化が必要であることや C や Java の一部の言語機能に制約が必要な理由は，回路設計とプログラミングのパラダイムに大きな開きがあることであると考えた．この違いによって大局的な並列性の解析や複雑なループの並列化など現在のコンパイラ技術でも困難な問題を引き起こし，ユーザによるコードの最適化を行う必要が生まれる．そこで本研究では，ディジタル回路設計と近いパラダイムを持ったプログラミングモデルである functional reactive programming (FRP)を用いたハードウェア設計ツール Mulvery を提案する．Mulvery は CPU と FPGA が共存するアーキテクチャにおいて利用されることを想定しフレームワーク化することでソフトウェアとハードウェアの協調設計を実現する．画像処理を例に評価を行ったところ，5x128 ピクセルの画像と 5x5 ピクセルのフィルタの畳み込み演算を 100MHz の動作周波数において 1 クロックで実行することができ，アーキテクチャ探索等を行うことなく高いスループットの回路を合成することを確認できた．また 32x32 個の LED マトリクスを用いた実験では，MCU のみでは LED マトリクスの制御さえ困難であったが，Mulvery はハードウェアオフロードによる LED マトリクス制御の高速化に加え，Ruby 言語の既存ライブラリなどを用いて Web サーバを記述することも可能で，ネットワーク経由での画像表示まで可能なものにした．

これらの応用に関する研究として，関数ポインタのサポートとクラウドコンピューティングへの FPGA の応用技術についての研究を行った．HLS においては，ポインタのように空間上の位置が動的に変化するようなプログラムを合成することは困難である．本研究で C 言語を用いる HLS ツールにおいてコードを抽象化し再利用性を高めるために重要な関数ポインタもこのような理由でサポートしコードの再利用性を高め，Mulvery のような FRP による高階関数を多用した抽象的なプログラミングモデルを実現するための研究に取り組んだ．クラウドコンピューティングの文脈では，Mulvery のデータフロー型の記述を生かし複数の FPGA で複数のタスクを分担して処理するための仕組みの基礎研究に取り組んだ．
反応閾値モデルを応用した自律分散システムによって，メンテナンス性の高さと高いフォールトトレランスを実現する手法を提案した．

（英訳）※和文要旨の場合(400 words)

The design of digital circuits for ASICs and FPGAs requires development in the register transfer level (RTL) written in a hardware description language (HDL).
Due to the nature of the syntax and the grammar, HDL is prone to bugs, and designers need various tools, skillful debugging techniques, and pieces of knowledge.
Therefore, a technology called high-level synthesis (HLS), a technology for synthesizing a circuit from a program written in a high-level language such as C/C++ or Java has already attracted attention.
Many HLS tools are available, and various programming languages have been applied.
However, there is a considerable gap between the description model of software development and circuit design.
The HLS tool may generate a complicated and colossal state machine that is difficult to automatic parallelization and pipelining since software programs lack information on data-flow and parallelism.
Hence the developer adds some information for optimization; otherwise, the throughput and the latency performance will be low.

Therefore, we have conducted this study to increase digital circuit design productivity on FPGA as the primary target by applying software programming techniques and methodology.