

# リアルタイム処理可能なデータ駆動型センサノードの構成法の検討

1255100 岡野 秀平 【 コンピュータ構成学研究室 】

## A Study on Data-Driven Sensor Nodes Capable of Real-Time Processing

1255100 Shuhei Okano 【 Advanced Computer Engineering Lab. 】

### 1 はじめに

近年、IoT(Internet of Things) 端末は様々な分野で活用され、それに伴い、高性能化が求められている。特に、スマートシティやスマート工場、自動運転技術に用いられるセンサノードには、長期稼働のための低消費電力化と安全な動作の保証のためのリアルタイム性の確保が要求される。

リアルタイム性を確保するためには、処理するタスクの優先度を考慮し、動的に資源を割り当てる必要がある。タスクの優先度に基づく動的スケジューリング方式のアルゴリズムとして EDF(Earliest Deadline First) がある。先行研究では EDF スケジューラを複数タスクの多重処理が可能なデータ駆動型プロセッサ DDP(Data-Driven Processor) に搭載する基礎的検討がなされている [1]。

本研究では、EDF スケジューラを搭載しセンサデータ処理が可能な DDP (以下 DDSH : Data-Driven Sensor Hub と呼ぶ) を Digilent 社の ZyboZ7-10 に組み込むことで、リアルタイムデータ駆動型センサノードの構築を目指す。そのため、汎用的なプログラムのタスクでも、DDSH 上でリアルタイム処理が可能な方式を提案する。

### 2 リアルタイムデータ駆動型センサノードの構成

リアルタイムデータ駆動型センサノードは、温度や流量のセンシングを行い、データ分析のための前処理の結果を中央制御端末へ送信する。ノード間の通信には BLE を用い、マルチホップ通信などのようにセンサノードを中継器としても利用することを想定している。

そのことから、図1の様に、リアルタイムにセンサハブ機能を実現する DDSH を FPGA 領域である PL 上に、各種センサや BLE とのインタフェースを制御する Cortex-A9 をプロセッサ領域である PS 上に実装する。

具体的には、取得したセンサデータを SPI/I2C と AXI を経由して組み込み用 OS である PetaLinux に送信し、DDSH 用のパッケージに成形する。一定量パッケージが成形されると、それをタスクとして DDSH に送信し、タスクを基に温度データの統計量を求め、UART を介した BLE 通信で制御用端末に統計量データの送信を行う。

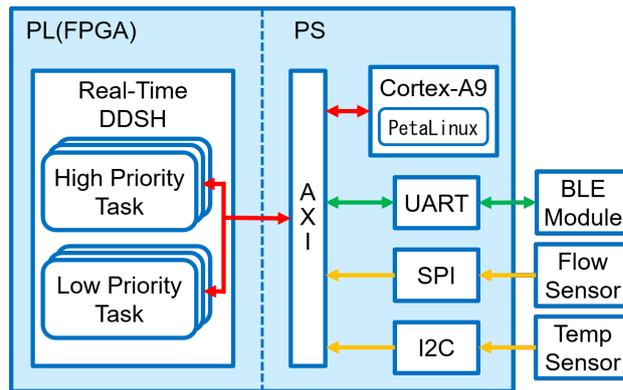


図1 リアルタイムデータ駆動型センサノードの構成

### 3 DDSH におけるリアルタイム処理方式

DDSH では、割り当てられたタスクの優先度に従ってリアルタイム処理を実現する必要がある。例えば、複数の統計量を算出する場合、重要な統計量の処理を優先的に行う。タスク数が DDSH の持つ多重処理能力を超える場合、超過分のタスクのパケットをキューに保管し、EDF スケジューリングにより優先度の高いパケットから処理が行われる。その際、DDSH の多重度は図2の LM(Load Monitor) で観測される。LM は DDSH 内のタスク数をパケット数から判断するため、2つのパケットを用いる二項演算や COPY によるパケットの複製を含むタスクでは正しい多重度の判定ができない。

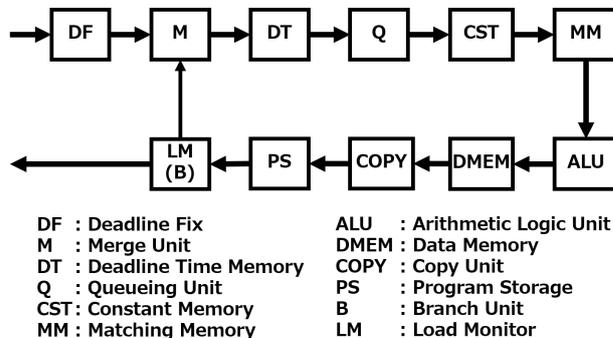


図2 DDSH の回路構成

### 3.1 多重実行タスクの観測方式

本研究ではLMで観測するパケットの粒度を粗くする観測方式を提案する。この方式では図3のように、一定範囲内の演算ノードの集合をモジュールとして扱い、タスク内の観測点を限定することで特定のパケットのモニタによりタスク数が観測できる。モジュールの出力から複数のパケットが観測される際は、LMで観測するパケットを観測点として設定し、他パケットを同期点として図2のMM(Matching Memory)で一時待機を行う。そうすることにより、従来の回路の構成をほとんど変更せずに二項演算などの命令が含まれた汎用タスクの多重度を観測することが可能になる。

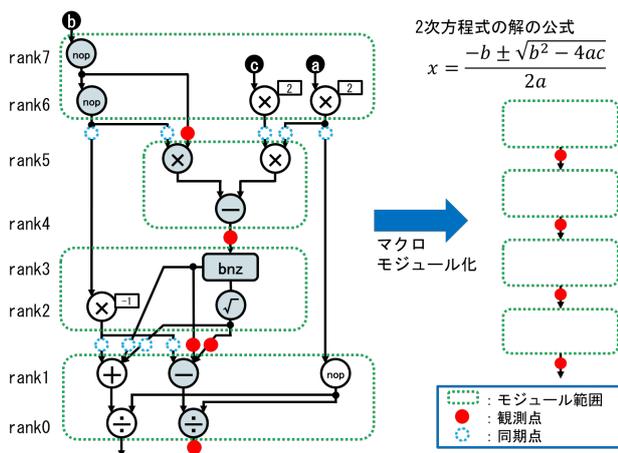


図3 プログラム(タスク)のマクロモジュール化

### 3.2 マクロモジュール化のアルゴリズム

タスクを観測し、状況に応じて停止させる際は、プログラム中で多くのノードとデータ依存関係がある経路上

#### Algorithm 1 $M \times M$ モジュール設定アルゴリズム

- 1: 他ノードとデータ依存関係が最も多いクリティカルパス  $CP$  を1つ選択
  - 2: 対象ノード  $T_N \leftarrow CP$  に含まれる  $rank0$  のノード
  - 3: **while**  $\exists$  未処理ノード **do**
  - 4:   **if**  $T_N \in CP$  **then**
  - 5:      $T_N$  から  $CP$  のノードを含むモジュールを作成
  - 6:   **else**
  - 7:      $T_N$  を含むモジュールを作成
  - 8:   **end if**
  - 9:   **for** 各モジュール **do**
  - 10:     同  $rank$  のノードをモジュールに組み入れる※
  - 11:   **end for**
  - 12:   Update  $T_N$ : 未処理ノードから最小  $rank$  ノードを選択※
  - 13: **end while**
- ※ノードから  $CP$  へのデータ依存関係を優先

の packets を観測することでタスク全体を停止しやすくなる。

そのことから、マクロモジュール化を行うにあたり、クリティカルパスを中心としたデータ依存関係を持つノードを優先的に対象としてモジュールを作成することで、粒度が粗くなることによるスケジューリング性能の低下を緩和することが可能である。

よって、Algorithm1 に示すモジュール設定アルゴリズムを定式化した。

また、条件分岐の出力を観測点に設定する場合、経路が True と False の2通りあるため、観測点も2か所設ける必要がある。このとき、1つのモジュールに対して2つの観測点が存在するが、出力パケットは片方の経路しか通らないため、LMでの観測も1か所のみとなり、他演算と同様に、タスクの実行停止/再開が実現できる。

## 4 評価

本研究で提案する観測方式を取り入れた DDSH をハードウェア記述言語である Verilog を用いて設計した。その後、DDSH 上で以下の典型的なタスクを実行した。

- ・1入力に対し、徐々に並列度が增加するタスク
- ・複数入力に対し、徐々に並列度が減少するタスク

時間制約が異なる上記タスクを複数実行した結果、いずれも正しく動作することを確認した。

また、本研究の提案方式を取り入れた DDSH と従来の EDF-DDP[2] を ZyboZ7-10 の PL 上に実装し、回路の規模を比較した結果、DDSH のレジスタ数が約 0.01% 増加した。このことから、タスクの観測方式を工夫することで既存の回路構成を微修正するだけで実用的なタスクの処理を実現する見通しが得られた。

## 5 まとめ

本研究では、センシングを用いたシステムを対象としたリアルタイム処理可能なデータ駆動型センサノードの構成法を検討した。

タスクをマクロモジュール化することでLMでタスク数を観測できるようになり、既存のスケジューリング回路の構成でも実用的なタスク処理の見通しを得られた。

また、今後の課題として、本提案方式を取り入れた DDSH のスケジューラビリティを確認する必要がある。そのためには、評価に適したベンチマーク・タスクセットについて検討する必要がある。

## 参考文献

- [1] K. Fukuda, et al., "Priority-Based Hardware Scheduler for Self-Timed Data-Driven Processor" PDPTA'17, pp.245-251, July 2017.
- [2] 岡野秀平, "データ駆動型プロセッサの EDF スケジューリング回路の最適化の検討", 高知工科大学修士学位論文, 2021.