

要旨

データ駆動型プロセッサにおける 優先度ベーススケジューリング方式の検討

福田 和馬

近年, IoT(Internet of Things) の普及に伴い, IoT 端末数が年々増加している. IoT 端末の中でも自転車やロボットなどのリアルタイムシステムでは制御対象の時間制約を守る必要があり, 個々のタスクの優先度を考慮して, どのタスクに資源を割り当てるかをスケジューリングする必要がある. しかし, ノイマン型逐次プロセッサでは, 実行するタスクを切り替える際にコンテキストスイッチオーバーヘッドが発生する. そこで, コンテキストスイッチオーバーヘッドを隠蔽できる並列処理プロセッサでのリアルタイム処理を実現するために, アルゴリズム自身が持つ本質的な並列性を表現できるデータ駆動計算モデルに着目する.

本研究では, データ駆動計算モデルに基いて動作するデータ駆動型プロセッサ (Data-Driven Processor : DDP) における優先度ベーススケジューリング方式を検討する. この方式として, タスク実行の開始/保留を DDP の入力部で制御する入力スケジューリング方式および, タスク実行の継続/中断を DDP 内で制御する内部スケジューリング方式を提案する. いずれの方式も, DDP 内のパケット流量を監視する多重度検知回路および, キューイング回路を DDP に搭載することで実現可能である.

提案スケジューリング方式を適用した DDP を 65nm CMOS 標準セルライブラリで合成し, 面積と遅延時間を評価した結果, どちらの方式も DDP のスループット性能にほとんど擾乱を与えずにスケジューリング制御が可能なことを確認した.

キーワード 優先度ベーススケジューリング, セルフタイム型パイプライン (STP), データ駆動型プロセッサ (DDP)

Abstract

A Study on Priority-Based Scheduling in Data-Driven Processor

Kazuma FUKUDA

In recent years, with the broad spread of Internet of Things (IoT) technology, the number of IoT devices is increasing year by year. Particularly in real time applications such as automobiles and robots the IoT device has to satisfy real-time constraints of various tasks. In order to do that, it is necessary to schedule execution resource allocation for individual tasks along with its priority. However, in the von Neumann sequential processor, a context switch for a prioritized task might lead performance degradation. Therefore, this study focuses on parallel processing processor that could hide such context switch overhead, especially, a data-driven model of computation representing intrinsic parallelism by a directed graph.

This paper proposes priority-based scheduling methods in a self-timed data-driven processor (DDP) which is a hardware implementation of a data driven model of computation. The one is an input scheduling method by which invocation or postpone of all task are controlled at input part of the DDP. The other is an internal scheduling method by which the stop or resume of the task execution are controlled within the DDP. Both methods can be introduced in the DDP by implementing a multiplicity detector, which observes the packet flow rate in the DDP, and priority-based task queues, which enqueue / dequeue tasks along with the detected multiplicity.

Using 65 nm CMOS library, we conducted the post-synthesis evaluation of the

DDP equipped with the proposed scheduling methods and then we measured actual delay time and area overhead of the additional scheduling circuits. As a result, we confirmed that both methods lead little overhead of throughput of the DDP.

key words Priority-Based Scheduling, Self-Timed Pipeline(STP), Data-Driven Processor(DDP)