Overview

Embedded Systems

- run multiple applications with real-time requirements such as throughput and latency.

Dataflow Modeling

- For design-time analyzability, application models, such as dataflow techniques, are used.

Homogeneous Synchronous Dataflow (HSDF)

- Actors connected through channels.
Multiprocessor System-on-Chip (MPSoc)

- architecture-aware model captures various system aspects.

Resource sharing

- MPSoC resources are shared to reduce cost.
- **Service guarantee** is the minimum service an arbiter guarantees each requestor.
Credit-Controlled Static-Priority (CCSP) Arbiter

- consists of a rate regulator and scheduler.
- schedules the highest-priority eligible requestor.
- the rate-regulator guarantees an allocated fraction of the resource.
Credit-Controlled Static-Priority (CCSP) Arbiter

- consists of a rate regulator and scheduler.
- schedules the highest-priority eligible requestor.
- the rate-regulator guarantees an allocated fraction of the resource.

Latency-rate ($L_R$) Service Guarantee

- is a linear model on the provided service.
Contributions

- A piece-wise linear service guarantee and its dataflow model.

Implications

- A given MPSoC resource can support more requestors, or
- Accommodate a given set of requestors with less resource capacity.
- Experimental result: savings from 26%-67% in memory bandwidth.
Outline

Piece-wise Linear Service Guarantee

Dataflow Model

Experimental Results
Service allocation for $r \in R$

- Abstract resource view: service unit and service cycle
- allocated service is the pair \((\text{burstiness}, \text{rate}) = (\sigma'_r, \rho'_r) \in \mathbb{R}^+ \times \mathbb{R}^+\).
- \[\sum_{r \in R} \rho'_r \leq 1 \text{ and } \sigma'_r \geq 1.\]

Service curves

- \(w\) - the requested service curve
- \(w'\) - the provided service curve
- backlog \(- \ q(t) = w(t) - w'(t)\)
- live line \(- \ \rho'_r \cdot (t - \tau_1 + 1)\)
- a requestor is live if \(w_r\) is above the live line.
Replenishment Policy

Active period

- is the maximum interval of time a requestor is backlogged and/or live.

Potential ($\pi_r(t)$)

- is the amount of budget a requestor has.
Service Guarantee

Service guarantee

- How do we compute the minimum guaranteed service for any active period?
  - by considering the active period with the maximum interference

\( LR \) Service Guarantee

- For every active period \([\tau_1, \tau_2]\), it guarantees a minimum service at a rate of \( \rho' \), after a maximum latency, \( \Theta \).

- However, it does not take into account bursty provided service; hence, it gives pessimistic WCRT.
Bi-rate Service Guarantee

- A higher rate ($\rho^*$) interval, followed by the regular service rate ($\rho'$).
- The higher rate, $\rho_r^* = 1 - \sum_{\forall s \in R_r^+} \rho'_s$. 

![Bi-rate Service Guarantee Diagram]

- live line
- $\cdot \hat{w}'$ (latency-rate)
- $\cdot \ddot{w}'$ (bi-rate)
Dataflow Model of Bi-rate Service Guarantee

- For $n$ requested service units within an active period,
- $s$ of them are served at the higher rate, and
- $n - s$ of them are served at the regular allocated rate.

CCSP dataflow model

Rate of provided service
Experiment Setup

Given

- a CCSP arbitrator SRAM memory controller,
- a video decoder application running on a GPP of a given frequency

We need to find

- the arbiter configuration that satisfies a throughput requirement,
  - both according to the latency-rate and the bi-rate service guarantees
Resource utilization

- E.g. At priority 3, where $\sigma = 1$, $\rho$ drops from 0.68 to 0.22. This is a memory bandwidth saving of 67%.
Pessimistic service guarantees lead to resource over-allocation.

A tight service guarantee enables
1. a given MPSoC resource to support more requestors, or
2. a given set of requestors to be accommodated with less resource.

We present a tight service guarantee for CCSP
1. a piece-wise linear guarantee for accurately capturing the provided service
2. a dataflow model for real-time timing analysis at system-level

Experiments show memory bandwidth savings from 26% - 67%.
Thank you! Questions?