

Materia:

**DISEÑO ELECTRÓNICO BASADO EN
SISTEMAS EMBEBIDOS**

Alumno:

Posadas Pérez Isaac Sayeg

Paniagua Rico Juan Julian

García Azzúa Jorge Roberto

Grado y grupo:

8°G

Profesor:

Garcia Ruiz Alejandro Humberto

Unidad 3 - Tarea 1:

Optimización global

Optimización global en sistemas embebidos

Introducción

En el campo de la ingeniería de sistemas embebidos, uno de los principales retos que se enfrenta es la necesidad de maximizar el rendimiento del sistema mientras se minimiza el consumo de recursos como energía, memoria y capacidad de procesamiento. Esta tensión entre eficiencia y funcionalidad ha llevado al desarrollo de múltiples técnicas de optimización, entre las que destaca la optimización global. A diferencia de la optimización local, que busca mejorar el desempeño de componentes individuales del sistema, la optimización global considera el sistema embebido como un todo integrado, abordando las interdependencias entre sus múltiples subsistemas para lograr un rendimiento óptimo a nivel general.

El concepto de optimización global en sistemas embebidos se refiere a un proceso en el que se intenta encontrar la mejor configuración posible de un conjunto de parámetros que afectan al sistema completo. Esto implica tomar decisiones sobre aspectos como el diseño del hardware, la asignación de tareas, el consumo energético, la planificación de recursos, la comunicación entre módulos, y las estrategias de ejecución en tiempo real.

Desarrollo

En sistemas embebidos, la optimización global implica una complejidad considerable, debido a que estos sistemas operan en entornos con restricciones rígidas y deben cumplir con requisitos específicos de rendimiento y fiabilidad. A menudo, los desarrolladores trabajan bajo limitaciones estrictas de tamaño, potencia y costo, lo que hace que las soluciones locales o parciales no sean suficientes para garantizar la calidad del sistema final.

Uno de los enfoques más comunes en la optimización global es la co-optimización hardware-software. Este paradigma consiste en diseñar simultáneamente tanto los componentes físicos como los algoritmos que se ejecutan sobre ellos, buscando un equilibrio que minimice el consumo de energía y maximice el rendimiento general.

Otro aspecto relevante es la planificación y asignación de tareas en sistemas embebidos de tiempo real. En estos sistemas, es fundamental que las tareas se ejecuten dentro de tiempos límites muy precisos (deadlines), lo que exige una gestión cuidadosa de los recursos de CPU, memoria y comunicaciones.

En sistemas distribuidos o con múltiples núcleos, la optimización global debe incluir mecanismos de paralelización y sincronización. No basta con asignar tareas a diferentes núcleos; también es necesario coordinar su ejecución para evitar cuellos de botella.

La optimización del consumo energético ha adquirido una importancia creciente. Muchos sistemas embebidos funcionan con baterías y deben operar durante largos períodos sin recarga. Las estrategias de optimización global incluyen técnicas como la gestión dinámica de voltaje y frecuencia (DVFS), el apagado selectivo de módulos inactivos y el uso de modos de bajo consumo.

Un área emergente en la optimización global es la integración de inteligencia artificial y aprendizaje automático. Mediante estas técnicas, es posible modelar el comportamiento del sistema y anticipar su carga de trabajo, permitiendo ajustes dinámicos y optimizaciones en tiempo real.

Ejemplos

- Vehículos autónomos: Se combinan múltiples sensores con potentes algoritmos de procesamiento en tiempo real para interpretar el entorno y tomar decisiones críticas. La optimización global balancea latencia, capacidad de procesamiento y consumo energético para garantizar seguridad y eficiencia.
- Dispositivos médicos implantables: Equipos como marcapasos deben ser extremadamente eficientes y confiables. La optimización global abarca desde

el diseño de circuitos de bajo consumo hasta algoritmos de control adaptativos.

- Sistemas IoT para monitoreo ambiental: Sensores remotos deben operar durante largos períodos de forma autónoma. La optimización global incluye el protocolo de comunicación, frecuencia de muestreo y estrategias de compresión de datos.
- Sistemas de control industrial distribuidos: Controladores embebidos en red coordinan tareas industriales. La optimización global gestiona recursos, sincronización y eficiencia energética para mejorar el rendimiento general.
- Dispositivos móviles con múltiples funciones: Teléfonos inteligentes ejecutan múltiples tareas. La optimización global distribuye procesos entre núcleos, gestiona el uso del GPU y adapta dinámicamente el sistema operativo.

Conclusión

La optimización global en sistemas embebidos representa una aproximación necesaria para el diseño eficiente de soluciones tecnológicas en entornos con restricciones críticas. A diferencia de la optimización local, este enfoque considera la interacción de múltiples componentes y capas del sistema, lo que permite alcanzar mejoras sustanciales en términos de rendimiento, consumo energético, fiabilidad y escalabilidad.

El desarrollo de técnicas avanzadas como la co-optimización hardware-software, la planificación inteligente de tareas, la gestión energética dinámica y la integración de algoritmos de inteligencia artificial está ampliando el horizonte de lo que es posible en términos de optimización global.

En resumen, la optimización global en sistemas embebidos no es simplemente una cuestión de mejora técnica puntual, sino una filosofía de diseño que reconoce la complejidad inherente de estos sistemas y busca soluciones que maximicen su valor funcional en contextos del mundo real.

Bibliografía

- ❖ 1. Wolf, W. (2012). *Computers as Components: Principles of Embedded Computing System Design*. Morgan Kaufmann.
- ❖ 2. Lee, E. A., & Seshia, S. A. (2017). *Introduction to Embedded Systems: A Cyber-Physical Systems Approach*. MIT Press.
- ❖ 3. Marwedel, P. (2011). *Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems*. Springer.
- ❖ 4. Buttazzo, G. C. (2011). *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*. Springer.
- ❖ 5. Kang, K.-D., Son, S. H., & Stankovic, J. A. (2003). Managing Deadline Miss Ratio and Overheads in Soft Real-Time Systems. *IEEE Transactions on Computers*.
- ❖ 6. Mittal, S. (2014). A Survey of Techniques for Improving Energy Efficiency in Embedded Computing Systems. *International Journal of Computer Aided Engineering and Technology*.
- ❖ 7. Givargis, T., & Vahid, F. (2002). Platune: A tuning framework for system-on-a-chip platforms. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*.
- ❖ 8. Benini, L., Bogliolo, A., & De Micheli, G. (2000). A Survey of Design Techniques for System-Level Dynamic Power Management. *IEEE Transactions on VLSI Systems*.