207577

บทคัดย่อ

ปัจจุบันนี้ได้มีการนำแพลตฟอร์มมัลติโพรเซสเซอร์มาใช้กับระบบเวลาจริงแบบเคร่ง (Hard Real-Time System) มากขึ้น เพื่อที่จะเพิ่มประสิทธิภาพให้กับระบบ อย่างไรก็ตามการพัฒนาในส่วนของความสามารถทนต่อความผิดพร่องบนระบบเวลาจริงแบบเคร่งที่อยู่บนแพลตฟอร์มมัลติโพรเซสเซอร์ (Hard Real-Time Multiprocessor Systems) ยังมีไม่มากนักเมื่อเทียบกับที่พัฒนาบนแพลตฟอร์มโพรเซสเซอร์เดี่ยว ดังนั้นวิทยานิพนธ์ฉบับนี้ต้องการนำเสนอวิธีการปรับปรุงเพื่อที่จะทำให้ระบบเวลาจริงแบบเคร่งบนแพลตฟอร์มมัลติโพรเซสเซอร์ มีความสามารถทนต่อความผิดพร่อง (Fault-Tolerant) ซึ่งสิ่งที่มีการปรับปรุงและพัฒนาขึ้นในงานวิจัยนี้แบ่งออกเป็นสามส่วน โดยส่วนแรกจะทำการประยุกต์วิธีการทำให้ระบบทนต่อความผิดพร่องที่ใช้กันบนโพรเซสเซอร์เดี่ยวมาใช้บนมัลติโพรเซสเซอร์ โดยใช้วิธีการจัดตารางการทำงาน แบบแบ่งส่วน (Partitioning Scheduling) ส่วนที่สองเป็นการพัฒนาและแก้ไขข้อบกพร่อง ของการจัดตารางการทำงานแบบครอบคลุม (Global Scheduling) ให้มีประสิทธิภาพดีขึ้น และส่วนสุดท้ายเป็นการปรับปรุงการจัดตารางแบบครอบคลุมที่พัฒนาในส่วนที่สองเพื่อให้ระบบนี้มีความสามารถทนต่อความผิดพร่อง ซึ่งผลลัพธ์จากการ Simulation ในงานวิจัยนี้ ทำให้ทราบถึงข้อดีข้อเสียของการจัดตารางแบบแบ่งส่วน และแบบครอบคลุม โดยการจัดตารางแบบครอบคลุมจะมีค่า Processor Utilization ดีกว่าแบบแบ่งส่วน แต่มีความซับซ้อนในการคำนวณมากว่า

207577

Abstract

Due to the increasing popularity and availability of chip-level multiprocessors, multiprocessor embedded systems have become more prevalent for a variety of applications. Some of these applications require an unusually high level of reliability, which may be achieved by integrating fault-tolerance capability to the underlying systems. For a hard real-time application, not only that the system continues to function correctly in the presence of fault but also the hard real-time requirement be fully satisfied. However, most fault-tolerant techniques for a hard real-time application have been developed for uniprocessor systems. Therefore, this thesis introduces novel techniques on the implementation of fault-tolerant, hard real-time multiprocessor system. First, we develop a baseline fault-tolerant, hard real-time multiprocessor scheduling by extending the existing fault-tolerant scheduling used in a uniprocessor system. To further improve the effectiveness of the scheduling process in a multiprocessor environment, we then devise a global fault-tolerant scheduling scheme where states of all the processors are considered for each task dispatched. Simulation results show that the global scheduling technique is able to better utilize each processing unit in the systems, resulting in a higher number of fault-tolerable tasks.