

멀티 코어 기반의 OFP 성능 향상 기법 연구

장현석* · 원현권** · 김인규** · 하석운*

*경상대학교 · **한국항공우주산업(주)

A Study of Performance Advanced Technique of the OFP on Multi-Core

Hyun-Seok Jang* · Hyeon-Kwon Won** · In-Gyu Kim** · Seok-Wun Ha*

*Gyeongsang National University · **Korea Aerospace Industries, Ltd.

E-mail : janghs@gnu.ac.kr

요 약

본 논문에서는 비행 운용 프로그램(Operational Flight Program, OFP)의 성능 향상을 위하여 멀티 코어 기법을 적용한 임무 컴퓨터(Mission Computer, MC)와 OFP의 디자인에 대해 기술하였다. 우선 멀티 코어 환경에서 태스크로 지정된 영역의 프로그램을 병렬 프로그램의 표준인 오픈엠프(OpenMp)를 사용하여 지정된 코어에서 제어하는 기법을 설계하여, 이를 적용한 멀티 코어 프로그램(Multi-Core Program, MCP)과 싱글 코어 프로그램(Single-Core Program, SCP)의 성능의 차이점에 대해 기술하였다. 항공기의 임무 컴퓨터 내에 멀티 코어를 지원하는 프로세서(General Processor Module, GPM)에 탑재되는 OFP중, 항법, 통신, 피아식별등의 비행 정보를 조종사에게 제공 및 제어하도록 설계된 전방 상향 제어 비행운용 프로그램(Integrated Up-Front Control OFP, IUFC OFP)의 성능 향상을 위한 멀티 코어 설계 기법을 제시 하였다.

ABSTRACT

In this paper, I present the design of Operational Flight Programs(OFPs) on a Multi-Core based Mission Computer(MC) for the optimized performance of the OFPs on Multi-Core based MC. The program assigned as tasks on Multi-Core environment can be scheduled by designing with the use of OpenMp, which is the standard for parallel programming. This paper also describes the differences between Multi-Core Program(MCP) on the technique and Single-Core Program(SCP) in terms of performance aspect. The new proposed design technique is applied to the Integrated Up-Front Control OFP(IUFC OFP) on General Processor Module where Multi-Core based. This paper describes the Multi-Core design technique for the optimized performance of the IUFC OFP, which display and control flight data(Navigation, Communication, Identification Friend or Foe) to pilot.

키워드

Operational Flight Program(비행 운용 프로그램), Mission Computer(임무 컴퓨터),

Integrated Up-Front Control(통합 전방 상향 제어기)

1. 서 론

임무 컴퓨터는 각종 항공 전자 장비와 1553B, Ethernet 등의 인터페이스를 통해 항법 시스템 및 무장 컴퓨터 제어 기능을 수행하고 IUFC, MFD 그리고 HUD를 통한 각종 비행 운용 데이터 입력, 비행정보를 시현 및 제어하는 항공 전자 시스템의 핵심 장비이다. 임무 컴퓨터에는 조종사에게 각종 비행정보를 제공 및 제어하는 비

행 운용 프로그램이 탑재되는데, 그 중 GPM 프로세서에 탑재되는 전방 상향 제어 비행운용 프로그램(Integrated Up-Front Control Operational Flight Program, IUFC OFP)은 통신, 항법, 피아식별등의 비행 정보를 시현하며, 조종사가 데이터를 제어할 수 있는 기능을 제공하는 IUFC를 제어하는 애플리케이션이다.

본 논문에서는 임무 컴퓨터 내 싱글 코어로 구성되어 있는 프로세서와 IUFC OFP를 멀티 코

이 기반으로의 설계 변경을 통하여 프로그램 성능을 향상시키는 기법을 제시하였다. 먼저 응용 프로그램에서 태스크로 지정된 영역을 병렬 프로그램 표준인 오픈엠프의 코어 지정 기법을 적용하여 태스크를 지정된 코어에서 제어하는 멀티 코어 프로그램 설계 방식을 사용함으로써, 싱글 코어 프로그램 방식과의 성능의 차이를 기술하였다. 이러한 설계 결과를 바탕으로 항공기에 장착 되는 임무 컴퓨터의 설계에 적용함으로써 IUFC OFP의 실행 성능을 최적화 하고자 하였다.

II. 싱글 코어 프로그램과 오픈엠프를 적용한 멀티 코어 프로그램의 성능 비교

병렬화를 구현하는 많은 기법들 중 본 논문에서는 오픈엠프를 적용한 멀티 코어 프로그램의 성능에 대해 기술하였다. 오픈엠프는 'OpenMP Architecture Review Board'라는 비영리 단체에서 편리하고 효율적으로 병렬 프로그램 구현할 수 있게 책정한 표준이다. 오픈엠프를 적용한 멀티 코어 프로그램의 성능 효과를 입증하기 위해 다중 태스크 처리 기반의 영상처리 옛지 검출 시뮬레이션 프로그램을 구현하였다. 싱글 코어 프로세서에서는 다중 태스크를 우선순위에 따라 순차적으로 실행하며, 멀티 코어 프로세서에서는 오픈엠프를 적용하여 태스크를 병렬적으로 코어에서 각각 제어하도록 구현하였다. 이 시뮬레이션 프로그램 실행 결과는 그림 1과 같다.

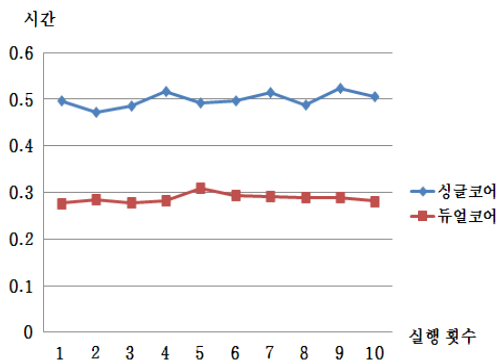


그림 1 싱글 코어 프로그램과 멀티 코어 프로그램 실행 시간

싱글 코어 프로세서에서 태스크를 순차적으로 실행한 시간은 약 0.49초이며, 듀얼 코어 프로세서에서 태스크를 병렬적으로 실행한 시간은 약 0.28초 이다. 오픈엠프를 적용한 멀티 코어 프로그램의 성능이 약 1.75배 효과적이다. 멀티 코어 프로그램의 성능의 효과 대해서 암달의 법칙에서 정의하고 있는 수식으로도 증명 할 수 있다. 암달의 법칙이란 시스템의 일부를 개선할 때 전

체적으로 얼마만큼의 최대 성능 향상이 있는지를 정의한다. 컴퓨터 프로그래밍에서는 하나의 응용 프로그램은 순차 영역과 병렬 영역으로 구현되고, 이를 합산한 결과가 프로그램의 성능으로 도출되는 것으로, 암달의 법칙을 수식화 하여 나타낸 성능 개선 효과는 다음과 같다.

- P : 개선 사항 부분, S : 성능 향상
- 성능 개선 효과 = $1 / (1 - P) + (P / S)$

본 논문에서 구현한 멀티 코어 프로그램을 암달의 법칙으로 수식화하면 응용 프로그램 내에서 멀티 코어 프로그램 영역이 P, 코어의 수가 S가 되며, 오픈엠프를 적용한 멀티 코어 프로그램의 최대 성능 향상은 다음과 같다.

- 싱글 코어 영역 : 18%, 멀티 코어 영역 : 82%
- 성능 개선 효과는 $1 / (1 - 0.82) + (0.82/2) = 1.7$ 이다.

암달의 법칙의 수식화에서도 싱글 코어 프로그램 대비 오픈엠프를 적용한 멀티 코어 프로그램의 성능이 효과적임을 알 수 있다.

III. 싱글 코어 기반의 IUFC OFP의 성능과 수행 시간

본 논문에서 기술한 IUFC OFP는 싱글 코어 프로세서에 탑재되어 동작되며 항공기 안전필수(Safety-Critical)요소를 만족하기 위한 시스템 요구도(System Requirement)를 기반으로 설계되었다. 또한 IUFC OFP는 25Hz 주기로 반복적으로 작업을 수행하고 20ms 시간내에 20~30% CPU 여유 처리 버퍼가 고려된 수행이 이루어져야한다. 그림 2에서는 현재 IUFC OFP의 태스크 수행 시간을 그래프로 나타낸 것으로 처음부터 다섯 번째까지 수행 주기에서는 허용된 수행 시간인 20ms 내에 작업 수행을 완료하였고, 이 주기에는 CPU 오버 타임(Over-Time)이 발생되지 않았으며 이는 IUFC OFP 작업이 정상적으로 완료됨을 의미한다. 하지만 여섯 번째 수행 주기부터 CPU 오버 타임이 발생하고 있는데 이는 작업이 정상적으로 수행되지 못했다는 것을 의미한다. CPU 오버 타임이 발생할 때 IUFC OFP 수행 시간은 평균적으로 약 22ms의 시간이 소요된다. IUFC OFP의 수행 시간을 태스크 별로 나누어 보면 UART 태스크의 수행 시간과 데이터 처리 태스크의 수행 시간으로 나눌 수 있다. 그림 2에서는 t1부터 t11까지 프로세서에서 생성되는 태스크를 나타내고 있다. IUFC OFP에서 생성하는 태스크 중 t5 태스크는 UART 통신을 담당하는 태스크 이고 t11 태스크는 데이터 처리를 담당하는 태스크다. UART 통신은 RS-485 시리얼 인터



그림 2 IUFCA OFF Tack 실행 시간

페이지로 초당 115200bps의 통신 속도를 제공한다. 수식으로 나타내면 통신시간을 T , Data(bit)의 크기를 S 라고 한다면 통신시간은 다음과 같다.

- T : 통신 시간(second), S : Data의 크기(bit)
- UART 통신시간 : $T = S / 115200$

이 수식을 UART 통신의 데이터 크기로 비교하면 다음과 같다. UART 통신 시간은 프로세서에서 IUFCA로 데이터를 송신하는 시간과 IUFCA로부터 데이터를 수신하는 시간으로 나뉘어진다. IUFCA로 송신하는 데이터의 크기는 184byte이며, IUFCA에서 수신하는 데이터 크기는 10byte이다. 데이터의 크기를 bit로 변환하면 각각 1472bit, 80bit가 된다.

- 데이터 송신 시간 : $T = 1472 / 115200$ 이며 $T = 0.012777\text{sec}$ 이다. 약 13ms 시간이 소요된다.
- 데이터 수신 시간 : $T = 80 / 115200$ 이며 $T = 0.000694\text{sec}$ 이다. 약 1ms 시간이 소요된다.

즉, UART 통신으로 14ms 시간의 소요되며 나머지 시간은 항전 장비와의 통신 및 데이터 처리에 소요된다. 하지만 수식으로 통신시간은 14ms 이지만 임무 컴퓨터의 환경적 요소와 프로세서에서 실행되는 우선순위의 작업 순서에 따라 실제 UART 통신시간을 측정해보면 약 16ms 시간이 소요된다. 나머지 약 5~6ms 시간은 시스템 관리, 항전 장비와의 통신과 데이터 처리에 소요된다. 본 논문에서는 이러한 오버 타이밍 현상

을 해결하기 위한 방법으로 듀얼 코어로 설계된 임무 컴퓨터의 프로세서와 멀티 코어 기법을 적용한 IUFCA OFF의 설계와 성능 향상에 대해 기술하였다.

IV. 멀티 코어 기법을 적용한 IUFCA OFF 설계 및 성능 향상 효과

듀얼 코어로 설계된 프로세서를 기반으로 멀티 코어 기법을 적용한 IUFCA OFF 구조는 그림 3와 같다.

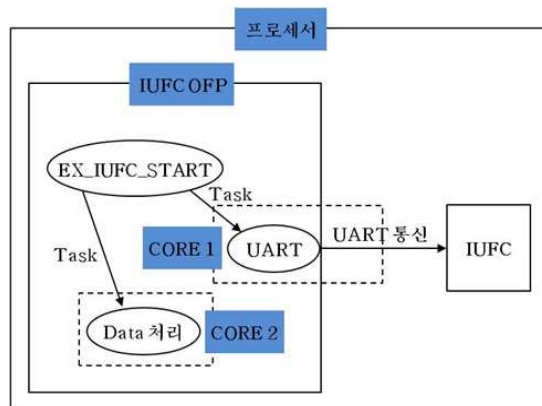


그림 3 듀얼 코어 기반의 IUFCA OFF 구조

듀얼 코어로 설계된 프로세서에서 각각의 코어는 코어1, 코어2로 명명하겠다. 듀얼 코어 기반의 IUFCA OFF를 싱글 코어 기반의 IUFCA OFF 설계 방식처럼 UART 통신과 데이터 처리를 담

당하는 두가지 태스크로 생성한다. 싱글 코어 기반의 IUFC OFP에서는 우선 순위에 따라 태스크의 작업이 순차적으로 수행되지만, 듀얼 코어 기반의 IUFC OFP에서는 UART 통신 태스크와 데이터 처리 태스크가 각각 코어1과 코어2에 독립적으로 지정되어 병렬적으로 수행된다. 즉, 싱글 코어 기반의 IUFC OFP에서는 우선순위로 나누어진 작업을 순차적으로 수행하여 20ms 내에 완료해야 하지만, 듀얼 코어 기반의 IUFC OFP는 코어1과 코어2가 병렬적으로 지정된 태스크의 작업을 각각 20ms 내에 실행됨을 의미한다. 싱글 코어 기반의 IUFC OFP와 듀얼 코어 기반의 OFP 작업의 실행 시간은 그림 4과 같다.

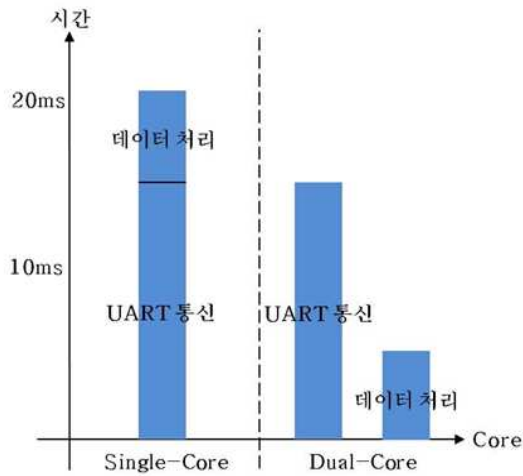


그림 4 IUFC OFP 수행 시간

이렇게 멀티 코어 기반의 IUFC OFP 설계 시 20ms내에 작업이 완료되어 오버 타임 현상을 해결할 수 있다. 또 다른 멀티 코어 기반의 IUFC OFP 설계 방식은 수행 시간별 코어 지정 방식이다. 예를 들어 싱글 코어 기반의 IUFC OFP와 같은 설계 방식처럼 태스크를 우선순위로 나누어서 순차적으로 수행하되 수행 시간이 많이 소요되는 작업 영역을 추출하여 멀티 코어 기반으로 설계 할 수 있다. 하지만 이 방법은 OFP 코드에서 수정할 코드를 찾아 그 영역을 일일이 멀티 코어 기반의 프로그램으로 수정해야 하는데, 이 경우 코드 수정 작업이 늘어나 프로젝트 개발 시간이 늘어나며, 추후 문제점이나 장비 추가와 같은 코드 변경사항 발생 시 멀티 코어 영역과 싱글 코어 영역을 같이 고려할 사항들이 발생할 수 있고, 코드 수정 작업에 어려움이 따를 수도 있다. 본 논문에서 제안하는 방식인 멀티 코어 환경에서 태스크로 선언된 영역을 각각의 코어에서 제어 및 처리하게 되면 코드 수정 사항 발생 시 더 적은 고려사항으로 쉽게 코드를 수정할 수 있을 것이다. 또한 프로세서가 쿼드 코어 기반으로 설계가 되었다면 UART 통신 태스크의 작업을 송신 영역과 수신 영역을 코어

별로 독립적으로 지정하여 병렬적으로 처리한다면 시간 단축의 성능 효과를 더 기대해 볼 수 있을 것이다.

V. 결 론

본 논문은 오픈엠프를 적용하여 싱글 코어 프로그램을 멀티 코어 프로그램으로 설계하여 병렬적으로 실행할 때 더 뛰어난 성능 효과를 입증하였다. 이를 통해 현재 싱글 코어 기반의 임무 컴퓨터와 IUFC OFP 설계 및 성능을 분석하고 임무 컴퓨터와 IUFC OFP를 멀티 코어 기반으로 설계함으로써 오버 타임 현상을 해결한 성능 효과 방안에 대해 기술하였다. IUFC OFP 뿐만 아니라, 다른 OFP(FC, HUD, MFD)에서도 멀티 코어 기반으로 설계 한다면 더욱 높은 성능의 효과를 기대해 볼 수 있을 것이다. 미래에는 뛰어난 성능의 첨단 항공기가 개발될 것이다. 이것은 더 많은 첨단 항진 장비들이 항공기에 장착되고 더 많은 데이터들이 항공기에서 운용되지만 더 짧은 시간에 더 효율적인 작업을 수행해야 한다는 의미이다. 멀티 코어 기반으로 설계 시 작업 시간의 단축이 충분히 예상 가능하며 더 많은 장비와 데이터가 늘어나도 주어진 시간 내에 충분히 임무를 수행 할 수 있을 것이다. 현재 멀티 코어 기반의 임무 컴퓨터 및 OFP 성능 향상과 신뢰성을 높일 수 있도록 지속적인 연구를 하고 있다.

참고문헌

- [1] 원현권 외, 실시간 임베디드 시스템용 자바 가상머신을 이용한 다기능 시현 비행 운용 프로그램 및 미들웨어 설계 및 구현, 한국항공우주학회지, 제39권 11호, 1060~1068, 2011.11
- [2] Eun-Ju Kim 외, Design of a digital cinema system by using RS485 communication module, MITA2011, 110~112, 2011.07
- [3] 정영훈 저, OpenMp 병렬 프로그래밍(멀티 코어를 위한 C, C++ 멀티스레딩의 표준), 프리렉, 2011.01
- [4] 정영훈 저, C, C++ 병렬 프로그래밍, 프리렉, 2010.07