

유비쿼터스 센서 노드를 위한 저전력 프로세서의 개발

Design of Ultra Low Power Processor for Ubiquitous Sensor Node

*신치훈, **오명훈, **박 경, **김성운

Chi-Hoon Shin, Myeong-Hoon Oh, Kyoung Park, Sung-Woon Kim

Abstract - In this paper we present a new-generation sensor network processor which is not optimized in circuit level, but in system architecture level. The new design build on a conventional processor architecture, improving the design by focusing on application oriented specification, ISA, and micro-architectural optimization that reduce overall design size and advance energy-per-instruction. The design employs harvard architecture, 8-bit data paths, and an compact 19 bit wide RISC ISA. The design also features a unique interrupt handler which offloads periodical monitoring jobs from the main part of CPU. Our most efficient design is capable of running at 300 KHz (0.3 MIPS) while consuming only about few pJ/instruction.

Key Words : Processor Architecture, Sensor Node, Low Power, Ubiquitous, Micro Architecture level Optimization

1. 서론 및 관련 연구

통신기술과 집적기술의 발달은 유비쿼터스 센서 네트워크를 탄생시켰다. 주변 환경에 센서노드들을 설치해 가상적인 세계와 물리적인 세계를 연결함으로써 유비쿼터스 센서 네트워크를 실현하는 것이다. 센서노드들은 자연 환경의 변화를 모니터링 하는 간단한 용도는 물론, 산업용 제어, 텔레메틱스, 홈오트메이션, 유비쿼터스 헬스케어 등의 다양한 용도에 걸쳐서 사용자에게 서비스를 제공하는 수단으로 활용될 수 있다 [2]. 또한 앞으로의 집적기술의 발달, 가격 하락 등의 요인으로 인해 앞으로 그 수요가 급증할 것으로 예상되고 있고 있다. 2006년 현재 노드 당 가격은 10 ~ 100 달러 선이지만 시장이 더 커져 안정화가 되는 2007 ~ 2010년에는 1 ~ 10달러 선에 진입할 것으로 예상되고 그 수요는 1억 3천 ~ 1억 8천 개에 이를 것으로 예상된다[5].

많은 수의 센서노드들을 일일이 유지 보수하는 것은 경제적이 못하다. 그러므로 센서노드들은 주로 일회용의 용도로 사용될 것이고 기본적으로 한정된 에너지를 효율적으로 사용해 장 시간동안 작동 가능하여야 할 것이다[1]. 에너지 효율성을 향상하기 위해서는 배터리의 성능을 향상시키는 것이 가장 간단한 해결책이나 배터리의 더딘 발전 속도를 고려하면 그와 같은 방법만으로는 문제를 해결할 수 없다. 일반적으로 센서노드 프로세서를 포함한 프로세서의 성능은 18개월에 두 배씩 증가하고 어플리케이션 알고리즘은 8.5개월에 두 배씩 증가한다고 볼 때 10년에 두 배씩 향상되는 배터리의 성능 향상 속도는 프로세서의 성능을 따라가지 못한다. 이런 이유로 장시간 작동 가능한 센서노드 설계에 있어 배터리 기술에 의존하기보다 시스템의 자체의 에너지 효율성을 확보하는 것이 무엇보다 중요한 이슈라고 할 수 있다[3].

전통적인 센서노드 플랫폼은 범용의 상용 저 전력 프로세서를 이용해 노드를 구성하고 그 위에 올라갈 OS를 센서 네트워크의 용도에 맞게 최적화 시키는 방식을 취하였다[3]. 그러나 이러한 방식으로는 프로세서 내부에서 센서네트워크의 용도에 불필요한 부분들의 에너지 소모를 막을 수 없다. 이 같은 한계를 극복하기 위해 단순한 기능을 가진 센서 네트워크 전용 프로세서들이 구현되기 시작했다[4]. 그리고 최근에는 단순히 프로세서의 기능을 줄이는 것을 뛰어 넘어 어플리케이션을 철저히 분석하고 아키텍처 레벨을 최적화함으로써 더 좋은 전력효율을 확보하려는 새로운 세대의 센서네트워크 프로세서들이 등장하고 있다[6, 7].

본 논문에서는 센서네트워크 어플리케이션들 중 모니터링 어플리케이션에 요구에 최적화된 프로세서를 제안한다. 2장에서는 모니터링 어플리케이션의 분석과 프로세서 사양을 결정하였고 3장에서는 인스트럭션 아키텍처의 디자인 4장에서는 마이크로 아키텍처 디자인을 다루었고 5장에서 결론과 향후 연구방안을 제시하며 끝맺는다.

2. 어플리케이션 분석과 Processor Specification 결정

2.1 어플리케이션 요구성능 분석

센서노드가 운영할 수 있는 다양한 어플리케이션 중 센서 네트워크의 보편적인 용도라고 할 수 있는 감시 (monitoring) 용도[7] 에 초점을 맞추어 프로세서를 최적화 하였다. 프로세서는 정해진 동작 (모니터링)을 일정한 간격으로 수행하여 외부로부터 데이터를 모으고 이것을 라디오에 전달하여 외부로 전송하게 한다. 표1[6]은 세부적인 용도에 따라서 요구하는 샘플링 (Sampling) 속도를 보여준다. [6, 7] 이 표에서 감시용 센서노드의 필요 성능은 실제로는 굉장히 낮음을 짐작할 수 있다. 샘플링 간격이 최소 몇 밀리 초에서 심지어 최장 몇 초까지 가능하기 때문이다.

* 과학기술연합대학교 컴퓨터 및 소프트웨어 공학과

** 한국전자통신연구원 서버플랫폼연구팀

대 상	샘플링속도 (Hz)
대 기 온 도	0.017 - 1
기 압	0.017 - 1
체 온	0.1 - 1
지 진 진 도	0.2 - 100
혈 압	50 - 100
엔 진 온 도 / 압 력	100 - 150
심 전 도	100 - 250

표 1. 측정대상별 샘플링속도

2.2 프로세서 사양 결정

본 논문에서 제안하는 센서노드 프로세서는 0.18um 공정에 0.5V의 Vdd, 300 KHz의 성능에 맞추어 설계되었다. 감시나 모니터링을 주로 하는 센서의 요구 성능은 수백 KHz 정도이다[6]. 커스텀 코어가 아닌 임베디드용 범용 프로세서를 사용하는 센서노드 플랫폼 프로젝트[3]의 경우 수십에서 수 MHz에서 수십 MHz의 성능을 가지므로 경우에 따라 필요성능의 수백 배를 초과하는 성능의 낭비가 초래된다.

인스트럭션 아키텍처로 8bit의 RISC 인스트럭션을 사용한다. 센서노드플랫폼에 대한 연구들은 대부분의 경우 8 비트나 16비트의 데이터 워드 사이클을 가진다[3, 6, 7]. 또한 CISC는 코드 사이클을 줄일 수 있어 상대적으로 RISC에 비해 전력이 덜 소모 되어 보일 수 있다. 그러나 실제에서는 CISC 인스트럭션을 처리하는 오버헤드 때문에 RISC보다 더 많은 전력을 소모한다. [7]

메모리는 인스트럭션용 메모리와 데이터 메모리를 구분하는 하버드 (harvard) 아키텍처로 구성되었다. 데이터 메모리의 사용이 많지 않아 데이터를 분리하여도 효율성의 문제가 없기 때문이다. 또한 인스트럭션 워드 폭과 데이터 워드의 폭을 다르게 구성할 수 있어 인스트럭션을 보다 자유롭게 최적화 할 수 있는 장점도 가지고 있다. 인스트럭션 메모리는 4K instruction 사이즈, 데이터 메모리는 256byte 로 구성한다. 복잡한 연산 기능을 배제하여 데이터량이 많지 않으므로 데이터 메모리는 상대적으로 작게 설계되었다.

본 논문에서 제안하는 프로세서를 이용한 센서노드 플랫폼은 다음과 같이 동작할 것이다. 타이머가 일정 간격으로 인터럽트를 발생시키고 이로 인해 깨어난 마이크로 프로세서는 외부환경으로부터 센서를 이용해 측정된 결과를 ADC (Analog to Digital Converter)를 통해서 읽어온다. 마이크로 프로세서는 결과를 패킷으로 만든 후 라디오 트랜시버에 전달하게 된다. 라디오는 패킷을 외부로 전송한다.

3. 인스트럭션 아키텍처의 구성

저 전력성이 좋은 ISA (Instruction Set Architecture) 는 최대한 단순함과 효율성을 강조하여 디자인 되어야 한다. 본 프로세서의 ISA 는 Peter J. Ashenden 가 제안한 Pico 프로세서[8]의 아키텍처를 변형하여 설계되었다. Pico Processor 는 아주 작고 효율적인 ISA를 가지고 있다.

표2는 인스트럭션의 구성을 보여준다. 총 15개의 인스트럭션으로 구성되어 있고 레지스터 메모리간의 동작방식에 따라 4가지로 구분된다. 각 구분 내에서 각각의 인스트럭션은 평선번호로써 다시 구분된다. 기본적인 연산기능과 메모리, 포

트 접근, 분기, 리턴 최소한의 기능으로 구성되었다. 그림 1은 인스트럭션의 포맷을 보여준다. 각 인스트럭션은 19비트의 길이를 가진다. 각 인스트럭션은 용도에 따른 구분 코드와 평선코드의 조합으로 구분할 수 있다.

구분	종류	동작	평선
ALU REG to REG, Const	ADD	ADD without carry in	001
	ADDC	Add with carry in	010
	SUB	Subtract without carry in	011
	SUBC	Subtract with carry in	100
	AND	Bitwise logical and	101
	OR	Bitwise logical inclusive or	110
Memory and I/O	XOR	Bitwise logical exclusive or	111
	LD	Load from memory	001
	ST	Store to memory	010
	INP	Input from port	011
Jump	OUT	Output to port	100
	BZ	Branch if zero	01
	JMP	Unconditional Jump	10
Stack & interrupt	RET	Return using stack	01
	RETI	Return from interrupt	10

표 2. 인스트럭션의 구성

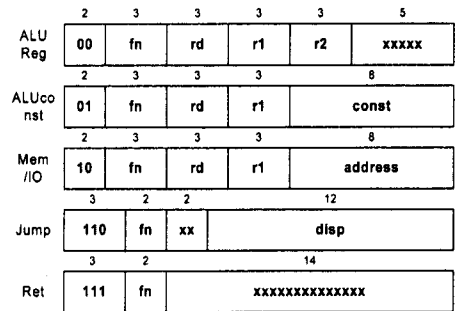


그림1. 인스트럭션 포맷

유념해야 할 것은 메모리/I/O 인스트럭션의 경우 어드레스는 데이터메모리의 절대번호를 의미하고 Jump 인스트럭션의 경우 인스트럭션메모리 상에서 상대번호 (다음 인스트럭션으로부터의 거리)를 의미한다는 것이다.

4. 마이크로 아키텍처 수준 최적화

그림2는 3장의 ISA를 마이크로 아키텍처 레벨로 디자인한 것이다. 기본적으로 외부 메모리가 없고 센서와 연결되는 I/O의 형태도 단순하므로 1 CPI (Cycle Per Instruction)의 싱글 사이클 머신으로 구현 하였다. 크게 데이터경로 (Data Path)와 컨트롤 경로 (Control Path) 그리고 이들을 컨트롤하는 상태 머신 (Finite State Machine) 으로 구성되어 있다.

프로세서는 인스트럭션 패치, 디코딩, 실행 등에 필요한 요소와 더불어 특별한 목적을 위한 하드웨어인 DIH (Direct Interrupt Handler)가 추가 되었다. 이 구조는 DMA (Direct Memory Access) 기술과 비슷하게 타이머, 센서데이터 읽기, 라디오로 보내기의 순서로 주기적으로 일어나는 정해진 동작을 프로세서의 주요 사이클을 거치지 않고 센서에서 라디오로 직접 전달하기 위해 고안되었다.

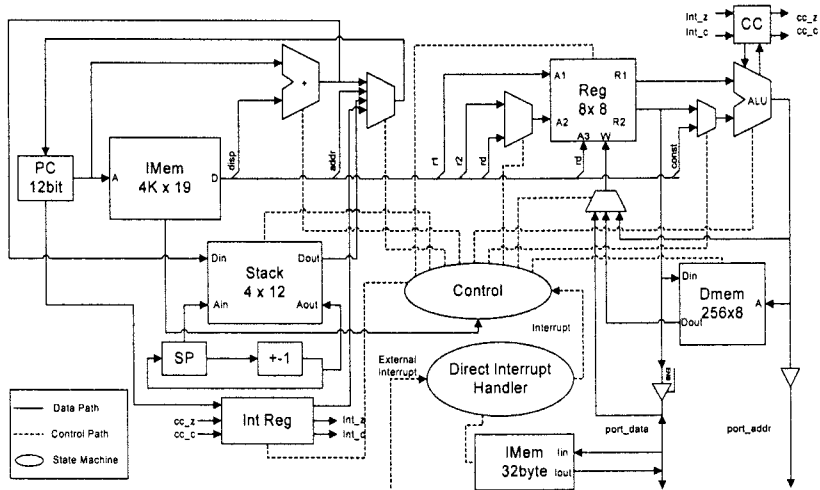


그림 2. 프로세서 마이크로 아키텍처 구성도

그림 2에서 볼 수 있듯이 DIH는 외부로부터의 모든 인터럽트를 먼저 받아 센서로부터 (혹은 타이머로부터) 오는 것 이면, 센서로부터 데이터를 읽어와 패킷사이즈의 전용 버퍼 에 저장하고 패킷화 한 후 라디오에 포트로 내보내게 된다. 만약 외부로부터의 인터럽트가 다른 곳으로부터 오는 것이라 면 DIH는 프로세서의 메인 컨트롤러에게 그 인터럽트를 보 낸다. 그림 3의 FSM은 DIH의 세부 동작을 보여준다. Idle 스테이트는 전력효율을 극대화하기 위해 존재한다.

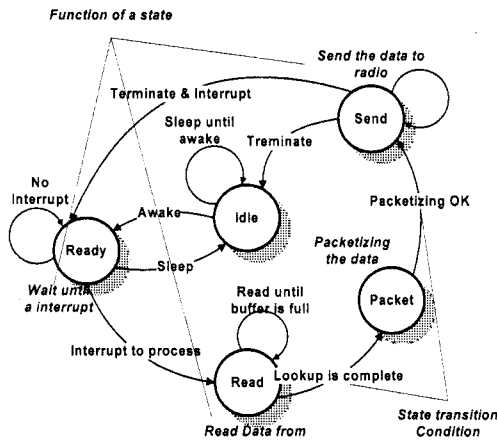


그림 3. Direct Interrupt Handler (DIH) FSM

5. 결론 및 향후 연구 방향

본 논문에서 센서 네트워크 어플리케이션의 특성에 적합한 프로세서를 설계, 구현해보았다. 현재 본 프로세서의 구현은 RTL 코드완료 후 시뮬레이션 단계에 있다. 특정 목적에 맞추어 설계된 일종의 ASIC (Application Specific Integrated Circuit)으로써 적절한 어플리케이션 상에서 측정한다면 인스트럭션 당 약 1pJ 정도의 전력 소비성을 보일 것으로 예상된다. 대략 100 pJ/instruction 정도의 성능을 가진 최신의 범

용 임베디드 프로세서들 보다 최대 100배 정도 성능향상이 가능함을 보여줄 수 있다. 센서노드는 제한적이고 일회적인 용도를 가졌을 뿐 아니라 최대한 비용을 낮추어야하는 특성을 가지고 있으므로 범용프로세서와 비교할 때 훨씬 매력적인 대안이 될 수 있다.

최근 발표된 논문에 따르면 그동안 센서 프로세서 연구에서 배제되어 왔던 파이프라인 분기예측 (Branch Speculation) 같은 프로세서 기법들을 적절히 이용하면 저전력성 향상에 큰 도움이 된다고 한다[7]. 또한 Clock으로 인한 전력누수를 원천적으로 막을 있는 비동기(Asynchronous) 기술도 프로세서 저전력성 향상에 도움을 줄 수 있다. 그러므로 향후 연구 과제로써 비동기 파이프라이닝, 비동기 DIH, 브랜치 예측이 적용된 프로세서를 구현할 것이다.

참 고 문 헌

- [1] J. Hill, et al, "System architecture directions for network sensors." ASPLOS 2000. Nov 2000.
- [2] C. Intanagonwiwat, et al. "Directed diffusion: a scalable and robust communication paradigm for sensor networks." MobiCom 2000.
- [3] J. Hill. System Architecture for Wireless Sensor Networks. PhD thesis, UC Berkeley, May 2003.
- [4] Virantha Ekanayake, et al. "An ultra low-power processor for sensor networks". ASPLOS 2004.
- [5] On World, Wireless Sensor Networks: Growing Markets, Accelerating Demand, July 2005.
- [6] Leyla Nazhandali, et al, "A Second-Generation Sensor Network Processor with Application-Driven Memory Optimizations and Out-of-Order Execution", ISCA 2006
- [7] Mark Hempstead, et al, "An Ultra Low Power system Architecture for Sensor Network Applications", ISCA 2006
- [8] Peter J. Ashenden "Pico Processor Architecture"