

에너지 효율적인 이미지 프로세서를 위한 Approximate Computing

I. 서론

오늘날 IT기기는 처리해야 할 데이터의 양은 점점 늘어나고 크기는 점점 작아지면서 높은 에너지 효율이 중요해지고 있다. 특히 <그림 1>에서 보듯이 웨어러블 디바이스의 사용량이 급증하면서 잦은 충전 문제가 핵심 이슈로 더욱 부상했다. 에너지 효율은 소비전력과 소모되는 시간의 곱으로, 에너지 효율을 향상시키기 위해서는 소비전력과 연산시간을 모두 감소시켜야 한다. 하지만 소비전력을 감소시키는 여러 가지 기법은 대부분 성능과 trade-off 관계를 갖고 있다. 소비전력을 감소시키기 위해서는 성능저하를 감수해야하고, 성능을 향상시키기 위해서는 소비전력의 증가를 감수해야 한다. 간단한 예로 공급전압의 크기를 줄이면 전력소모가 매우 감소되지만, 그만큼 성능이 저하된다.

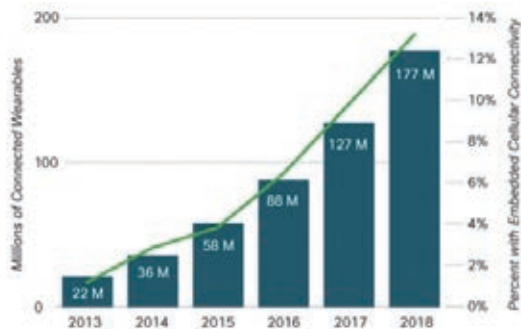
이러한 문제를 해결할 수 있는 기법 중 하나가 approximate computing이다. Approximate computing은 정확도를 줄임으로써 전력소모의 감소와 성능 향상을 동시에 얻을 수 있다. 특히 사람의 감각과 관련된 어플리케이션에서는 약간의 정확도를 손실해도 사



김영민
광운대학교
컴퓨터정보공학과



김성현
광운대학교 컴퓨터공학과



<그림 1> 세계 웨어러블 디바이스 시장 전망
(자료원: Cisco VNI Mobile)

람이 인지하지 못할 정도인 경우 영향이 없기 때문에, approximate computing을 적용하기 적합하다. 이처럼 approximate computing은 에너지 효율을 위한 설계에서 매우 중요하고 유용한 방법이다.

II. Approximate Computing

그동안 컴퓨터 플랫폼은 정확한 연산을 수행해야 한다는 것이 당연시 됐다. 이런 조건에서의 기법은 대부분 성능과 소비전력 사이의 trade-off를 수반한다. 하지만 부정확한 연산을 허용하게 되면 위의 trade-off관계를 극복할 수 있다.

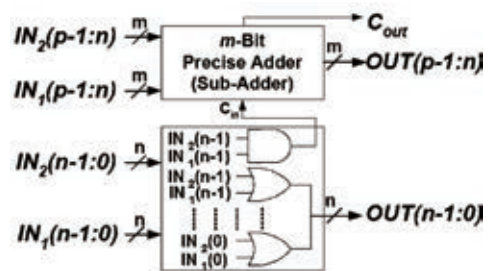
Approximate computing은 연산의 정확도를 어느 정도 허용할 만큼 손실을 감안하고 연산의 단순화와 소비전력 감소라는 두 가지 이득을 얻을 수 있는 방법이다. 특히 <그림 2>의 이미지 처리, 영상 처리, 음성인식 같이 사람의 감각과 관련 된 어플리케이션에서는 눈과 귀로 인지할 수 없는 작은 오류를 무시함으로써 에너지 효율과 성능의 향상을 기대할 수 있다.

Approximate computing은 기본적으로 산술연산 회로에 적용이 가능하다. 우선 산술연산의 기본이 되는 adder에 가장 먼저 적용됐다. Adder에 적용할 때는 1-bit adder 자체의 트랜지스터 구조를 변형하는 방법들이 많이 이용됐다. Yang^[1]은 트랜지스터 레벨에서

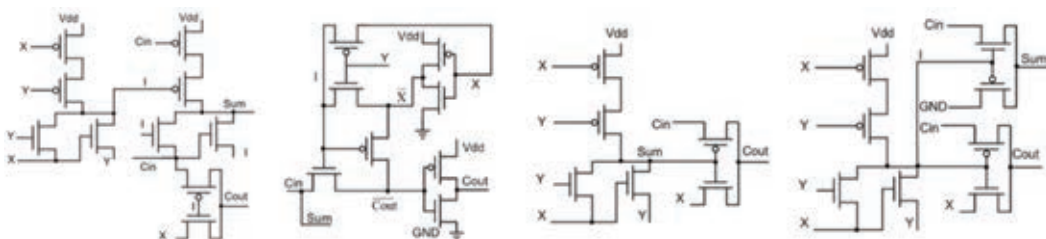


<그림 2> approximate computing 적용에 적합한 어플리케이션의 종류

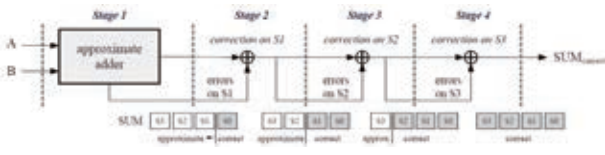
XOR/XNOR 게이트 구조와 패스트랜지스터(PTL)를 기반으로 한 1-bit approximate adder들을 연구하였다 (<그림 3> 참조). 위 adder들은 정확한 연산을 하는 1-bit full adder와 비교하여 소비전력과 트랜지스터 수의 감소(즉, adder의 크기 감소)는 물론, 회로 딜레이의 감소도 보여주었다. 이렇게 원래의 덧셈연산보다 조금 부정확한 연산을 수행하도록 하면서 소비전력 감소와 성능 향상을 시킨 새로운 구조의 adder들이 많이 제안됐다. 1-bit adder뿐만 아니라 multi-bit adder도 많은 연구가 진행됐다. multi-bit중 하위비트일수록 값이 매우 작다는 점을 이용하여 일정 LSB(Least Significant Bit)만을 approximate adder로 대체하고 나머지 비트는 정확한 adder를 이용하여 이득을 가져왔다. 대표적으로 Mahdiani^[2]가 제안한 LSB부분이 OR게이트로 이루어진 adder가 있다(<그림 4> 참조). 이 adder는 간단한 구조로 인해 power, delay, area에서 모두 이득을 가져왔지만 오류의 확률이 너무 높게 나오는 경향을 보였고, 관련 연구에서 많은 참조가 되어왔다. 이 외에도 approximate adder에 대한 많은 연구가 진행됐는데, 그 중에는 adder 자체의 특성에 초점을 맞춘 것이 아닌 실제 어플리케이션의 동작에 초점을 맞춘 연구도 나왔다. 동작중인 어플리케이션에서는 실시간으로 필요한 정확도가 바뀔 수 있는



<그림 4> OR게이트로 이루어진 approximate adder^[2]



<그림 3> 트랜지스터 구성이 다른 여러 가지 1-bit approximate adder^[1]



〈그림 5〉 4단계로 파이프라인 된 approximate adder^[3]

데, 이러한 특성을 이용하여 필요에 따라 정확도를 조절할 수 있는 accuracy-configurable adder가 제안되기도 했다^[3]. 이 adder는 정확한 계산을 위한 파이프라인 단계를 나누어 정확도에 따라 단계를 조절할 수 있도록 한 것이다(〈그림 5〉 참조).

Approximate computing은 multiplier에도 적용됐다. Multiplier 자체의 cell 구조를 바꿔서 approximation하거나, 곱셈 연산의 특성을 이용하여 연산되는 bit수를 줄여서 approximation하는 방법들이 연구됐다. 이 구현들에 대한 자세한 내용은 다음 절에서 설명하도록 한다.

현재는 divider까지 approximate computing 연구가 확장되었다. divider는 크기가 매우 크고, 전력을 아주 많이 소비하기 때문에 높은 소비전력 감소의 효과가 기대되는 logic이다. Multiplier와 마찬가지로, divider에 사용되는 subtracter를 대체하는 등 cell 구조를 변형하는 방

법과 나눗셈의 특성을 이용하여 연산되는 bit수를 줄이는 방법 등이 연구되고 있다. 이것 역시 다음 절에서 자세히 살펴보도록 한다.

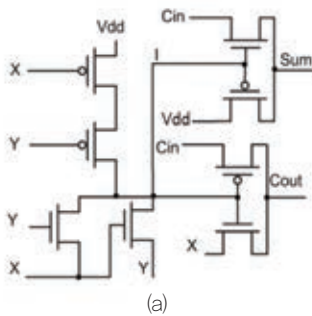
Approximate computing이 적용된 logic은 전력 효율을 높이기 위한 목적이긴 하지만, 부정확한 연산이 추가되므로 신뢰성을 평가하기 위한 측정 기준이 필요하다. Liang^[4]은 이를 위한 새로운 성능 지표를 제안했다. ED(Error Distance), MED(Mean ED)와 NED(Normalized ED) 이렇게 세 가지 지표가 있는데, ED는 오류가 있는 출력과 올바른 출력 간의 산술 차이로 정의되고, MED는 ED의 평균 그리고 NED는 해당 logic에서 나올 수 있는 최대 출력 값과 ED의 비율로 정규화한 ED이다. 이 지표들은 approximate computing 관련 연구에서 유효성과 신뢰성을 평가하는데 사용되고 있다.

이 절까지는 approximate computing의 필요성 및 중요성, 원리, 그리고 연구 동향을 알아봤다. 이제 다음 절에서는 몇 가지의 예를 들어 그 구체적인 구현방법에 대해 살펴보겠다.

Ⅲ. Approximate Computing 구현 방법

1. Approximate Adder

덧셈은 산술연산의 기본이며, adder는 산술연산 logic의 가장 기본이 되는 logic이다. [5]에서는 XNOR 구조를 기반으로 full adder에서 셀 구조를 변형한 1-bit approximate adder를 제안했다(〈그림 6(a)〉 참조). Accurate full adder와 비교하여 트랜지스터 수가 20% 감소되어 면적과 소비전력 부분에서 이득을 얻는다. 정확도의 경우 〈그림 6(b)〉의 진리표를 참조하면 확인할 수 있다. 총 8번의 input case중 2번의 경우 error가 발생하는 것을 볼 수 있는데, error의 확률 자체도 중요하지만 한 가지 더 중요한 것은 error가 발생하는 input case가 적절하냐는 것이다. Error case는 Cin이 1인 경우에 한정해서 나타나는 것이 더 유리하다. 올림수가 발생할 확률보다 발생하지 않을 확률이 더 크기 때문이며, 이것은 1-bit approximate adder를 multi-bit으로 확장하

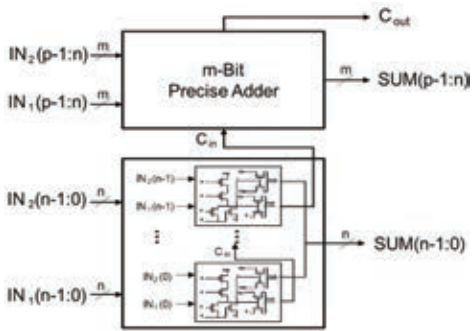


(a)

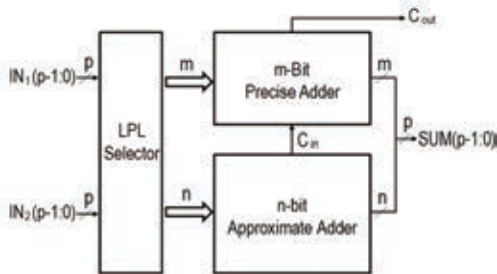
X	Y	Cin	Accurate adder		Approximate adder		
			Cout	Sum	Cout	Sum	ED
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	0	1	0	1	0
0	1	1	1	0	1	1	1
1	0	0	0	1	0	1	0
1	0	1	1	0	1	1	1
1	1	0	1	0	1	0	0
1	1	1	1	1	1	1	0

(b)

〈그림 6〉 1-bit approximate adder의 구조(a), accurate adder와 (a)의 진리표(b)^[5]



〈그림 7〉 LSB가 1-bit approximate adder로 구성된 multi-bit approximate adder^[5]



〈그림 8〉 입력 패턴에 맞춰 정확도 조절이 가능한 approximate adder^[6]

여 비교함으로써 검증할 수 있다. 〈그림 7〉은 〈그림 6(a)〉를 이용하여 만든 multi-bit approximate adder이다. 이 adder는 이전 연구에서 나온 adder들 보다 더 좋은 error rate, MED, NED를 보여준다. 어플리케이션에 따라 approximate adder로 구성되는 LSB 수를 정하여 정확도와 성능의 trade-off를 결정할 수도 있다.

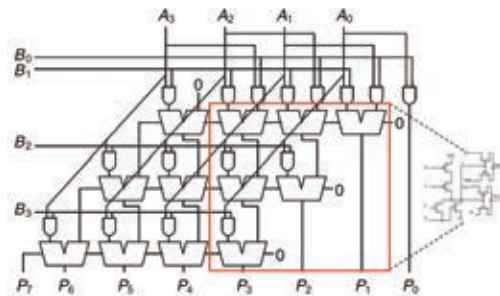
위 아이디어에서 실시간 동작에 대한 부분까지 고려하여 제안한 adder가 〈그림 8〉이다^[6]. 어플리케이션의 동작 중에는 실시간으로 다른 데이터가 입력될 것인데, 입력 패턴을 확인하여 error의 발생 여부를 어느 정도 예측하여 approximate adder의 비트 수를 결정한다는 아이디어이다. ‘LPL Selector’ 모듈이 실시간 입력 패턴에 따라 하위비트 수를 조절해주는 역할이다. 이 모듈에서 입력마다 error의 발생 범위를 계산하여 하위 n-bit을 결정해 주기 때문에 다른 approximate adder들에 비해 매우 좋은 error metric(error rate, MED, NED)을 보여준다.

2. Approximate Multiplier

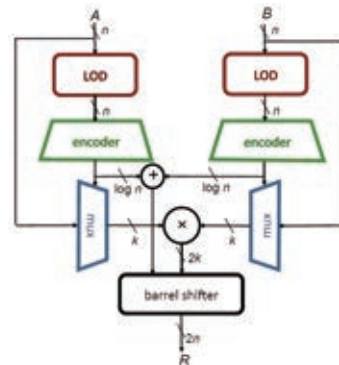
곱셈 또한 어플리케이션에서 필수적인 산술 연산이다. Approximate multiplier는 정확한 연산을 하는 내부의

adder를 approximate adder로 대체하여 만든 구조이다. 〈그림 9〉에서와 같이 4x4 multiplier의 경우, 하위 3-bit의 결과를 만들 때 사용되는 adder를 approximate adder로 대체하는 것을 보여준다. 정확도와 성능의 trade-off 필요에 맞게 approximation시킬 하위 비트 수를 결정하여 사용한다. 16x16 multiplier로 가정했을 때 원래의 multiplier와 비교하여 10~40%의 전력 감소, 6~16%의 면적 감소, 22~40%의 delay 감소 효과를 보였다.

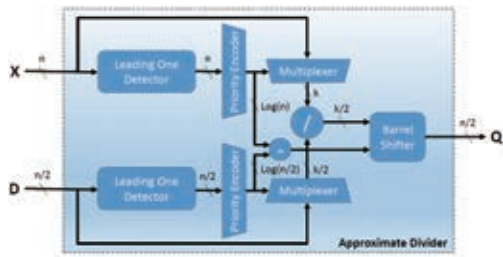
곱셈의 특성을 이용하여 approximation을 시킬 수도 있다. Hashemi^[7]는 하위비트 자리의 연산이 결과 값에 큰 영향을 미치지 않는 곱셈의 특성을 이용하여 곱셈연산의 bit수를 감소시키는 구조를 제안했다(〈그림 10〉 참조). 이 구조는 최상위비트에서부터 가장 처음으로 1이 나오는 자리를 찾은 후 k-bit만큼 남기고 나머지 하위 비트는 버리는 구조이다. 이렇게 k-bit로 감소된 input을 받아 conventional multiplier에 적용하여 곱셈을 한다. 이 approximate multiplier는 약 58%의 절전 효과를 보여준다.



〈그림 9〉 내부 adder를 approximate adder로 대체한 approximate multiplier



〈그림 10〉 Input data의 bit수를 감소시켜 연산하는 approximate multiplier^[7]



〈그림 11〉 Input data의 bit수를 감소시켜 연산하는 approximate divider^[8]

3. Approximate Divider

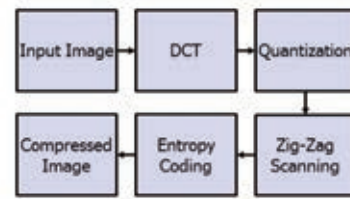
현재는 divider까지도 활발하게 연구되고 있는 상황이다. Divider는 매우 큰 logic이기 때문에 approximation 하여 어플리케이션에 적용할 경우 성능 면에서 매우 큰 이득을 가져 올 수 있다. Hashemi^[8]는 앞선 multiplier에서 사용했던 특성을 divider에도 적용하여 approximate divider를 제안했다. LOD(Leading One Detector)에서 가장 처음으로 1이 나오는 자리를 찾고, mux를 통해 사용될 bit자리를 정하여 conventional divider로 입력해 준다. 마지막에는 shifter를 통해 자리 수를 복원해 준다. 값이 더 작아지는 나눗셈의 특성 상, 연산 bit수를 줄여서 발생하는 오차 범위에 대해 3.08%의 뛰어난 성능을 보였다. 또한 최대 70%의 절전 효과를 보여줬다.

지금까지 approximate computing을 적용한 산술연산 logic에 대하여 알아보았다. 그렇다면 실제 어플리케이션에 적용이 가능한 것인지 다음 절에서 예시를 통해 살펴보겠다.

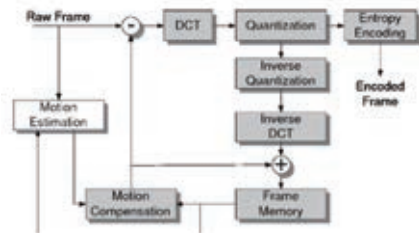
IV. Approximate Computing의 응용 예

Approximate computing 기법은 사람의 감각이 사용되는 application이라면 어디든 용이하게 적용될 수 있다. 우리는 이중에서도 눈의 인지능력과 관련된 응용프로그램을 살펴보기로 한다.

〈그림 12〉는 이미지 압축 표준인 jpeg과 영상 압축 표준인 mpeg의 encoder 구조이다. Gupta^[9]는 제안한 approximate adder의 효율성을 검증하기 위해 jpeg encoder와 mpeg encoder에 적용시켜 압축된 결과와 전력감소를 보여줬다. Approximate adder는 〈그림 12〉의 DCT (Discrete Cosine Transform)와 Inverse DCT모듈에 적용됐다. DCT는 덧셈연산을 많이 수행하며 손실이

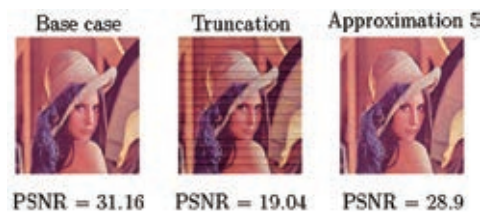


(a)



(b)

〈그림 12〉 jpeg encoder 구조(a), mpeg encoder 구조(b)^[9]



〈그림 13〉 각 case별 output quality^[9]

일어나도 되는 과정이기 때문에 적용이 가능한 것이다. 〈그림 13〉은 원래의 jpeg결과와 truncation기법을 이용한 결과 그리고 approximation된 jpeg의 결과를 비교한 것이다. Approximation 기법이 이용된 결과는 원래의 결과와 육안으로 거의 차이가 없는 것을 확인할 수 있다. 이때 소비전력의 감소는 약 52%로 매우 좋은 결과를 보여준다. MPEG의 경우도 영상 압축 결과가 육안으로 차이가 나지 않는 수준에서, 약 42%의 전력 감소를 보여줬다.

이 절에서 알아본 것처럼, 사람의 인지능력이 사용되는 어플리케이션에는 approximate computing방법이 적용될 수 있다. 실제 어플리케이션에 적용 할 때는 많은 테스트를 통하여 어느 정도 approximation이 허용되는지 검증을 거친 후 적용하면, 요구되는 정확도와 성능의 trade-off를 맞출 수 있을 것으로 기대한다.

V. 결론

지금 현재 반도체 관련 학계나 기업에서는 반도체의 성



능과 집적도를 높이기 위해서 “무어의 법칙”으로 대변되는 회로선폭의 축소를 위해 매진하고 있다. 하지만, 소비전력이 높아서 기존 설계기술을 활용한 고성능 SoC로는 미래의 웨어러블 기기에 적용하기 불가능하다. Approximate computing 기법은 정확도와 소비전력간의 trade-off관계를 갖기 때문에 웨어러블 SoC에 적용되기 적절하다. 꼭 웨어러블 기기가 아니더라도 앞으로의 트렌드가 크기는 점점 작아지지만 에너지 효율은 높은 기기를 원하기 때문에 이 조건에 맞추어 최적화를 진행할 수 있을 것으로 기대되는 방법이다.

본고에서 살펴본 approximate logic들과 이것이 적용된 응용프로그램들을 볼 때 충분히 상용 가능하며, 필요로 하는 성능을 달성할 수 있을 것으로 기대할 수 있다. Approximate computing의 정확도와 성능 trade-off의 연구는 앞으로도 활발하게 진행될 것으로 보이며, 최적화된 저전력 이미지 프로세서 설계 등에 다양하게 이용될 수 있을 것으로 예상된다.

참고 문헌

- [1] Z. Yang, et al., “Approximate xor/xnor-based adders for inexact computing”, in Proc. Nanotechnology (IEEE-NANO), pp. 690–693, 2013.
- [2] H.R. Mahdiani, et al., “Bio-inspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications”, IEEE Trans. on Circuits and Systems I: Regular Papers, vol. 57, no. 4, pp. 850–862, 2010.
- [3] A.B. Kahng and S. Kang, “Accuracy-configurable adder for approximate arithmetic designs”, in Proc. DAC, pp. 820–825, 2012.
- [4] J. Liang, J. Han and F. Lombardi, “New metrics for the reliability of approximate and probabilistic adders”, IEEE Trans. on Computer, vol. 62, no. 9, pp. 1760–1771, 2013.
- [5] S. Kim and Y. Kim, “Energy-efficient hybrid adder design by using inexact lower bits adder”, in Proc. APCCAS, pp. 355–357, 2016.
- [6] S. Kim and Y. Kim, “Adaptive Approximate Adder (A3) to Reduce Error Distance for Image Processor”, in Proc. ISOC, pp. 295–296, 2016.
- [7] S. Hashemi, R. I. Bahar and S. Reda, “DRUM: A Dynamic Range Unbiased Multiplier for Approximate Applications”, in Proc. ICCAD, pp. 418–425, 2015.
- [8] S. Hashemi, R. I. Bahar and S. Reda, “A Low-Power Dynamic Divider for Approximate Applications”, in Proc. DAC, 2016.
- [9] V. Gupta, et al., “Low-power digital signal processing using approximate adders”, IEEE Trans. on CAD of Integrated Circuits and Systems, vol. 32, no. 1, pp. 124–137, 2013.



김영민

- 1999년 8월 연세대학교 전자공학과 학사
- 2003년 4월 미시건 대학교 EECS 석사
- 2007년 12월 미시건 대학교 EECS 박사
- 1999년 8월~2000년 10월 삼성전기 연구원
- 2007년 10월~2009년 7월 미국 Qualcomm 연구원
- 2009년 8월~2015년 2월 울산과학기술원(UNIST), 조교수
- 2015년 3월~현재 광운대학교 컴퓨터정보공학부, 부교수

<관심분야>
SoC 설계 및 설계자동화(CAD)



김성현

- 2016년 2월 광운대학교 컴퓨터공학과 학사
- 2016년 3월~현재 광운대학교 컴퓨터공학과 석사과정

<관심분야>
SoC 설계 및 설계자동화(CAD), 임베디드 시스템 설계, 저전력 SoC 설계