



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2024-0078193
(43) 공개일자 2024년06월03일

- | | |
|---|--|
| <p>(51) 국제특허분류(Int. Cl.)
G06F 7/533 (2006.01) G06F 9/30 (2018.01)</p> <p>(52) CPC특허분류
G06F 7/5334 (2013.01)
G06F 9/30029 (2013.01)</p> <p>(21) 출원번호 10-2022-0160911</p> <p>(22) 출원일자 2022년11월25일
심사청구일자 2022년11월25일</p> | <p>(71) 출원인
포항공과대학교 산학협력단
경상북도 포항시 남구 청암로 77 (지곡동)</p> <p>(72) 발명자
이영주
경상북도 포항시 남구 청암로 77
박군호
경상북도 포항시 남구 청암로 77
김영석
경상북도 포항시 남구 청암로 77</p> <p>(74) 대리인
특허법인(유한)아이시스</p> |
|---|--|

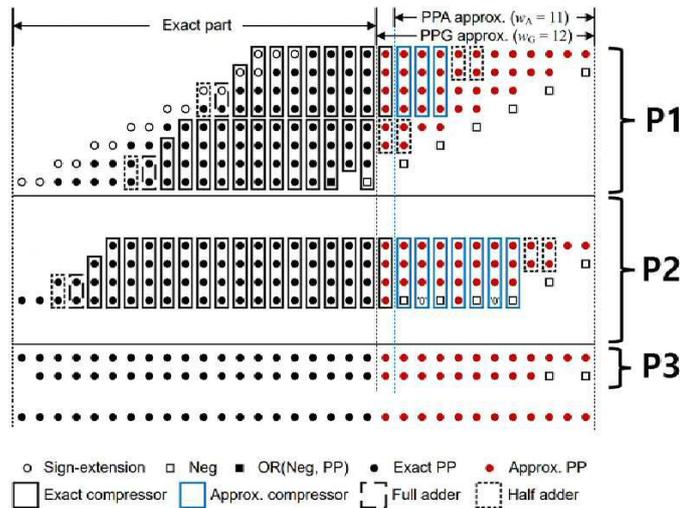
전체 청구항 수 : 총 17 항

(54) 발명의 명칭 부분곱 생성기, 부분곱 누적기 및 이들을 포함하는 근사 곱셈기

(57) 요약

본 실시예는 승수(multiplier)와 피승수(multiplicand)를 입력받고 에러 집중화된 부분곱을 생성하는 부분곱 생성기(partial product generator)로, 상기 부분곱 생성기는 상기 승수와 상기 피승수를 논리 연산하여 에러 집중화된 부분곱을 형성하되, 상기 논리 연산은 상기 승수와 상기 피승수로 논리합(OR) 연산하여 수행한다.

대표도 - 도1



이 발명을 지원한 국가연구개발사업

과제고유번호 1711174849
 과제번호 2022R1A2C2092521
 부처명 과학기술정보통신부
 과제관리(전문)기관명 한국연구재단
 연구사업명 개인기초연구(과기정통부)
 연구과제명 차세대 초고효율 6G 베이스밴드 시스템을 위한 알고리즘-하드웨어 융합최적화
 기여율 40/100
 과제수행기관명 포항공과대학교
 연구기간 2022.09.01 ~ 2023.02.28

이 발명을 지원한 국가연구개발사업

과제고유번호 1711160341
 과제번호 2020-0-01461-003
 부처명 과학기술정보통신부
 과제관리(전문)기관명 정보통신기획평가원
 연구사업명 정보통신방송혁신인재양성
 연구과제명 지능형 의료영상 진단 솔루션 개발
 기여율 30/100
 과제수행기관명 아주대학교산학협력단
 연구기간 2022.01.01 ~ 2022.12.31

이 발명을 지원한 국가연구개발사업

과제고유번호 1711159913
 과제번호 2020-0-01612-003
 부처명 과학기술정보통신부
 과제관리(전문)기관명 정보통신기획평가원
 연구사업명 정보통신방송혁신인재양성
 연구과제명 Grand ICT연구센터(금오공과대학교)
 기여율 30/100
 과제수행기관명 금오공과대학교 산학협력단
 연구기간 2022.01.01 ~ 2022.12.31

명세서

청구범위

청구항 1

승수(multiplier)와 피승수(multiplicand)를 입력받고 에러 집중화된 부분곱을 생성하는 부분곱 생성기(partial product generator)로,

상기 부분곱 생성기는 상기 승수와 상기 피승수를 논리 연산하여 에러 집중화된 부분곱을 형성하되,

상기 논리 연산은 상기 승수와 상기 피승수로 논리합(OR) 연산하여 수행하는 부분곱 생성기.

청구항 2

제1항에 있어서,

상기 부분곱 생성기는,

정확한 부분곱과 오차 있는 부분곱을 생성하되,

상기 오차 있는 부분곱은 상기 정확한 부분곱에 비하여 더 큰 부분곱과 더 작은 부분곱을 포함하며,

상기 부분곱 생성기는 상기 더 작은 부분곱을 생성하는 경우보다 더 큰 부분곱을 생성하는 경우가 더 많은 부분곱 생성기.

청구항 3

제1항에 있어서,

상기 논리합 연산은,

OR 게이트로 수행하는 부분곱 생성기.

청구항 4

제1항에 있어서,

상기 부분곱 생성기는,

연속된 피승수(multiplicand) 두 비트 중 더 큰 자리 한 비트와

연속된 승수(multiplier) 세 비트 중 제일 큰 자리 한 비트를 논리합 연산하여 부분곱을 생성하는 부분곱 생성기.

청구항 5

제4항에 있어서,

상기 부분곱 생성기는,

상기 연속된 피승수 두 비트 중 더 큰 자리 한 비트가 일 입력으로 제공되고,

상기 연속된 승수 세 비트 중 제일 큰 자리 한 비트가 타 입력으로 제공되는 OR 게이트를 포함하는 부분곱 생성기.

청구항 6

제1항에 있어서,

상기 부분곱 생성기는,

두 비트의 피승수 $a_j a_{j-1}$ 중 a_j 와

세 비트의 승수 $b_{2i+1}b_{2i}b_{2i-1}$ 중 b_{2i+1} 를 논리합 연산하여 부분곱을 생성하는 부분곱 생성기. (i, j : 0 이상의 정수, 단, a_{-1} 와 b_{-1} 은 모두 0이다.)

청구항 7

제1항에 있어서,

상기 부분곱 생성기는,

상기 a_j 가 일 입력으로 제공되고,

상기 b_{2i+1} 가 타 입력으로 제공되는 OR 게이트를 포함하는 부분곱 생성기. (i, j : 0 이상의 정수, 단, a_{-1} 와 b_{-1} 은 모두 0이다.)

청구항 8

복수의 부분곱들을 입력받고 여러 집중화된 합을 생성하는 부분곱 누적기로,

상기 부분곱 누적기는,

상기 부분곱들에서 각각 한 비트 씩을 제공받고 두 비트를 논리곱(AND) 연산하여 출력하고 및 한 비트를 동일한 논리값을 출력하여 상기 부분곱에 포함된 “1”의 개수를 출력하되 연산 오차가 집중화된 합을 출력하는 부분곱 누적기.

청구항 9

제8항에 있어서,

상기 부분곱 누적기는,

정확한 합과 오차 있는 합을 생성하되,

상기 오차 있는 합은 상기 정확한 합에 비하여 더 작은 값인 부분곱 누적기.

청구항 10

제8항에 있어서,

상기 부분곱 누적기는,

입력된 상기 복수의 부분곱의 한 비트가 출력되고,

입력된 상기 복수의 부분곱의 다른 두 비트를 논리곱(BND) 연산하는 AND 게이트를 포함하는 부분곱 누적기.

청구항 11

제8항에 있어서,

상기 부분곱 누적기는,

셋 이상의 부분곱들 별로 한 비트씩 입력받고,

상기 셋 이상의 부분곱들에서 동일한 인덱스에 위치하는 어느 한 비트를 그대로 출력하고,

다른 두 비트를 논리곱 연산하여 출력하는 AND게이트를 포함하는 부분곱 누적기.

청구항 12

제11항에 있어서,

상기 부분곱 누적기는,

다른 하나의 부분곱의 한 비트가 더 입력되고,

상기 다른 하나의 부분곱의 한 비트는 연산되지 않는 부분곱 누적기.

청구항 13

제8항에 있어서,
 상기 부분곱 누적기는,
 서로 다른 세 개의 부분곱에서 동일한 인덱스에 위치하는 $p_{4i,j}$, $p_{4i+3,j}$, $p_{4i+2,j}$ 비트들을 제공받고,
 $p_{4i,j}$ 의 논리값을 그대로 출력하고,
 $p_{4i+3,j}$, $p_{4i+2,j}$ 의 논리값에 대하여 논리곱(AND) 연산하는 부분곱 누적기.
 (i, j: 0 이상의 정수)

청구항 14

제13항에 있어서,
 상기 부분곱 누적기는,
 다른 하나의 부분곱에서 상기 동일한 인덱스에 위치하는 $p_{4i+1,j}$ 비트를 제공받되,
 상기 $p_{4i+1,j}$ 비트는 무시하는 부분곱 누적기.(i, j: 0 이상의 정수)

청구항 15

근사 곱셈기로, 상기 근사 곱셈기는,
 승수(multiplier)와 피승수(multiplicand)를 입력받고 여러 집중화된 부분곱을 생성하는 부분곱 생성기(partial product generator); 및
 상기 복수의 부분곱들을 입력받고 여러 집중화된 합을 생성하는 부분곱 누적기를 포함하며,
 상기 부분곱 생성기가 생성하는 상기 부분곱은 예러가 정확한 연산 결과에 비하여 큰 방향으로 집중화되고,
 상기 부분곱 누적기가 생성하는 상기 합은 예러가 정확한 연산 결과에 비하여 작은 방향으로 집중화된 것인 근사 곱셈기.

청구항 16

제15항에 있어서,
 상기 부분곱 생성기는 상기 승수와 상기 피승수로 논리합(OR) 연산하여 수행하는 상기 부분곱을 연산하는 근사 곱셈기.

청구항 17

제15항에 있어서,
 상기 부분합 누적기는,
 상기 부분곱들에서 각각 한 비트 씩을 제공받고 두 비트를 논리곱(AND) 연산하여 출력하고 및 한 비트를 동일한 논리값을 출력하여 상기 부분곱에 포함된 “1”의 개수를 출력하되 연산 오차가 집중화된 합을 출력하는 근사 곱셈기.

발명의 설명

기술 분야

본 개시는 일반적으로 부분곱 생성기, 부분곱 누적기 및 이들을 포함하는 근사 곱셈기와 관련된다.

[0001]

배경 기술

- [0002] 뉴럴 네트워크의 연산은 입력과 회소 행렬로 이루어진 가중치를 곱하여 형성된 부분곱을 누적하는 MAC(multiply and accumulate) 연산을 주요한 요소로 한다. MAC 연산을 연산 시간, 소모되는 에너지 및 면적 효율적으로 수행하기 위한 근사 곱셈의 개념은 비단 뉴럴 네트워크 뿐만 아니라 DSP(디지털 신호 처리) 시스템의 에너지 효율성을 개선하는 데 중요한 역할을 하는 것으로 간주되었다.
- [0003] 제한된 전력/에너지 예산 하에서 단순화된 하드웨어 설계로 인한 부정확한 결과를 허용함으로써 디지털 필터링, 멀티미디어 처리 및 기계와 같은 일련의 곱셈을 사용하여 전력을 많이 소비하지만 오류 복원력이 있는 신호 처리 알고리즘을 수행할 수 있다.

발명의 내용

해결하려는 과제

- [0004] MAC 연산은 크게 곱셈과 누적의 두 가지 요소로 나눌 수 있다. 부분합의 개수를 감소시킴으로써 곱셈 및 누적의 연산을 간단히 할 수 있는 알고리즘으로는 부스 알고리즘(Booth's algorithm)이 있다. 부스 알고리즘은 곱셈의 대상인 승수(multiplier)의 비트 시퀀스에 따라 0 , $+1 \times$ 피승수(multiplcand), $-1 \times$ 피승수(multiplcand), $+2 \times$ 피승수(multiplcand), $-2 \times$ 피승수(multiplcand) 및 0 로 부분곱(paratial product)을 형성하여 이들을 누적한다.
- [0005] 그러나, 이러한 부스 알고리즘에 의한다 하더라도 정확한 곱셈 결과를 얻을 수 있는 연산 장치를 형성하기 위하여 필요한 면적과 이들이 소모하는 전력은 무시할 수 없다. 따라서 다소간의 부정확한 결과를 감수하더라도 면적 및 에너지 효율적인 곱셈 연산 장치를 형성할 필요성이 증대되고 있다.
- [0006] 본 기술로 해결하고자 하는 과제 중 하나는 상술한 종래 기술의 필요성을 해소하는 것이다. 즉, 본 기술로 해결하고자 하는 과제 중 하나는 면적 및 에너지 효율적인 근사 곱셈 연산 장치를 제공하기 위한 것이다.

과제의 해결 수단

- [0007] 본 실시예는 승수(multiplier)와 피승수(multiplicand)를 입력받고 예러 집중화된 부분곱을 생성하는 부분곱 생성기(partial product generator)로, 상기 부분곱 생성기는 상기 승수와 상기 피승수를 논리 연산하여 예러 집중화된 부분곱을 형성하되, 상기 논리 연산은 상기 승수와 상기 피승수로 논리합(OR) 연산하여 수행한다.
- [0008] 본 실시예의 어느 한 측면에 의하면, 상기 부분곱 생성기는, 정확한 부분곱과 오차 있는 부분곱을 생성하되, 상기 오차 있는 부분곱은 상기 정확한 부분곱에 비하여 더 큰 부분곱과 더 작은 부분곱을 포함하며, 상기 부분곱 생성기는 상기 더 작은 부분곱을 생성하는 경우보다 더 큰 부분곱을 생성하는 경우가 더 많다.
- [0009] 본 실시예의 어느 한 측면에 의하면, 상기 논리합 연산은, OR 게이트로 수행한다.
- [0010] 본 실시예의 어느 한 측면에 의하면, 상기 부분곱 생성기는, 연속된 피승수(multiplicand) 두 비트 중 더 큰 자리 한 비트와 연속된 승수(multiplier) 세 비트 중 제일 큰 자리 한 비트를 논리합 연산하여 부분곱을 생성한다.
- [0011] 본 실시예의 어느 한 측면에 의하면, 상기 부분곱 생성기는, 상기 연속된 피승수 두 비트 중 더 큰 자리 한 비트가 일 입력으로 제공되고, 상기 연속된 승수 세 비트 중 제일 큰 자리 한 비트가 타 입력으로 제공되는 OR 게이트를 포함한다.
- [0012] 본 실시예의 어느 한 측면에 의하면, 상기 부분곱 생성기는, 두 비트의 피승수 $a_j a_{j-1}$ 중 a_j 와 세 비트의 승수 $b_{2i+1} b_{2i} b_{2i-1}$ 중 b_{2i+1} 를 논리합 연산하여 부분곱을 생성한다. (i, j : 0 이상의 정수, 단, $a-1$ 와 $b-1$ 은 모두 0이다.)
- [0013] 본 실시예의 어느 한 측면에 의하면, 상기 부분곱 생성기는, 상기 a_j 가 일 입력으로 제공되고, 상기 b_{2i+1} 가 타 입력으로 제공되는 OR 게이트를 포함한다. (i, j : 0 이상의 정수, 단, $a-1$ 와 $b-1$ 은 모두 0이다.)
- [0014] 본 실시예는 복수의 부분곱들을 입력받고 예러 집중화된 합을 생성하는 부분곱 누적기로, 상기 부분곱 누적기는, 상기 부분곱들에서 각각 한 비트 씩을 제공받고 두 비트를 논리곱(AND) 연산하여 출력하고 및 한 비트를 동일한 논리값을 출력하여 상기 부분곱에 포함된 "1"의 개수를 출력하되 연산 오차가 집중화된 합을 출

력한다.

- [0015] 본 실시예의 어느 한 측면에 의하면, 상기 부분곱 누적기는, 정확한 합과 오차 있는 합을 생성하되, 상기 오차 있는 합은 상기 정확한 합에 비하여 더 작은 값이다.
- [0016] 본 실시예의 어느 한 측면에 의하면, 상기 부분곱 누적기는, 입력된 상기 복수의 부분곱의 한 비트가 출력되고, 입력된 상기 복수의 부분곱의 다른 두 비트를 논리곱(BND) 연산하는 AND 게이트를 포함한다.
- [0017] 본 실시예의 어느 한 측면에 의하면, 상기 부분곱 누적기는, 셋 이상의 부분곱들 별로 한 비트씩 입력받고, 상기 셋 이상의 부분곱들에서 동일한 인덱스에 위치하는 어느 한 비트를 그대로 출력하고, 다른 두 비트를 논리곱 연산하여 출력하는 AND게이트를 포함한다.
- [0018] 본 실시예의 어느 한 측면에 의하면, 상기 부분곱 누적기는, 다른 하나의 부분곱의 한 비트가 더 입력되고, 상기 다른 하나의 부분곱의 한 비트는 연산되지 않는다.
- [0019] 본 실시예의 어느 한 측면에 의하면, 상기 부분곱 누적기는, 서로 다른 세 개의 부분곱에서 동일한 인덱스에 위치하는 $p_{4i,j}$, $p_{4i+3,j}$, $p_{4i+2,j}$ 비트들을 제공받고, $p_{4i,j}$ 의 논리값을 그대로 출력하고, $p_{4i+3,j}$, $p_{4i+2,j}$ 의 논리값에 대하여 논리곱(AND) 연산한다. (i, j : 0 이상의 정수)
- [0020] 본 실시예의 어느 한 측면에 의하면, 상기 부분곱 누적기는, 다른 하나의 부분곱에서 상기 동일한 인덱스에 위치하는 $p_{4i+1,j}$ 비트를 제공받되, 상기 $p_{4i+1,j}$ 비트는 무시한다. (i, j : 0 이상의 정수)
- [0021] 본 실시예는 근사 곱셈기로, 상기 근사 곱셈기는, 승수(multiplier)와 피승수(multiplicand)를 입력받고 에러 집중화된 부분곱을 생성하는 부분곱 생성기(partial product generator); 및 상기 복수의 부분곱들을 입력받고 에러 집중화된 합을 생성하는 부분곱 누적기를 포함하며, 상기 부분곱 생성기가 생성하는 상기 부분곱은 에러가 정확한 연산 결과에 비하여 큰 방향으로 집중화되고, 상기 부분곱 누적기가 생성하는 상기 합은 에러가 정확한 연산 결과에 비하여 작은 방향으로 집중화된 것이다.
- [0022] 본 실시예의 어느 한 측면에 의하면, 상기 부분곱 생성기는 상기 승수와 상기 피승수로 논리합(OR) 연산하여 수행하는 상기 부분곱을 연산한다.
- [0023] 본 실시예의 어느 한 측면에 의하면, 상기 부분합 누적기는, 상기 부분곱들에서 각각 한 비트 씩을 제공받고 두 비트를 논리곱(AND) 연산하여 출력하고 및 한 비트를 동일한 논리값을 출력하여 상기 부분곱에 포함된 “1”의 개수를 출력하되 연산 오차가 집중화된 합을 출력한다.

발명의 효과

- [0024] 본 실시예에 의하면 보다 작은 면적과 낮은 전력 소모량이 낮은 부분곱 생성기 및 부분곱 누적기가 제공된다.

도면의 간단한 설명

- [0025] 도 1은 본 실시예에 의한 연산 장치로 근사 곱셈을 수행하는 과정을 예시한 도면이다.
- 도 2(a)는 표 2로 예시된 종래 기술에 의한 부스 인코더의 게이트 레벨 회로도이고, 도 2(b)는 본 실시예에 의한 부분곱 생성기를 게이트 레벨로 도시한 회로도이다.
- 도 3(a)는 종래 기술에 의한 부스 인코더(booth encoder)로 십진수 13에 상응하는 이진수 01101과 2에 상응하는 이진수 10과의 연산을 예시하는 도면이다.
- 도 3(b)는 본 실시예에 의한 부분곱 생성기로 동일한 승수와 피승수를 대상으로 부분곱의 연산하는 과정을 예시하는 도면이다.
- 도 4(a)는 종래 기술에 의한 부분곱 누적기의 게이트 레벨 회로도이고, 도 4(b)는 본 실시예에 의한 부분곱 누적기의 게이트 레벨 회로도이다.
- 도 5(a)와 도 5(b)는 각각 본 실시예에 의한 부분 곱 생성기와 부분곱 누적기에서 근사 비트 수에 따른 오차 분포를 도시한 도면이다.
- 도 6은 본 실시예에 의한 부분 곱 생성기와 부분곱 누적기를 동시에 사용하여 곱셈기를 구성하였을 때의 최종 오차를 도시한 도면이다.

도 7(a) 및 도 7(b)는 각각 FIR filtering과 CNN기반 이미지 분류에 본 실시예에 의한 부분 곱 생성기를 적용할 때의 정확도와 에너지 효율을 나타낸 도면이다.

발명을 실시하기 위한 구체적인 내용

- [0026] 이하에서는 첨부된 도면들을 참조하여 본 실시예를 설명한다. 도 1은 본 실시예에 의한 연산 장치로 근사 곱셈을 수행하는 과정을 예시한 도면이다. 도 1을 참조하면, 적색점은 에러 집중화된 부분곱을 생성하는 부분곱 생성기(partial product generator)에 의한 결과 비트이고, 흑색점은 정확한 곱셈에 의한 결과 비트를 나타내며, 채워지지 않은 흑색 원은 부호 비트(sign bit)를 나타낸다. 도시된 바와 같이 부분합들은 부호 비트를 포함하고, 모두 16 비트의 근사 곱셈에 의한 결과 비트 및 정확한 곱셈의 결과 비트를 포함하는 것으로 예시되었다. 또한, 부분합의 말단에 위치하는 채워지지 않은 흑색 사각형은 2의 보수(2's complement)를 생성하기 위하여 더해지는 값이다.
- [0027] 도 1로 예시된 것과 같이 곱셈 연산은 복수의 부분곱(partial product)들을 연산하고, 연산된 부분곱들을 합산하여 수행된다. 도 1에서 각각의 부분곱은 동일한 행으로 표시되었다. MSB(most significant bit)에 가까운 비트일수록 곱셈 연산 결과에 더 많은 영향을 미친다. 따라서, MSB에 가까운 비트들에 대하여는 정확한 부분곱을 생성하고, 생성된 정확한 부분곱들을 정확하게 합산한다.
- [0028] 도 1으로 예시된 곱셈 연산은 피승수와 승수 모두 16 비트이고, 연산 결과는 32비트인 예를 도시한다. 이 중에서 하위 12 비트는 근사 연산을 수행하여도 연산 결과에 미치는 영향이 상대적으로 작아 본 실시예에 의한 부분곱 생성기와 부분곱 누적기를 이용하여 연산을 수행하여도 무방하다. 다만, 상위 20 비트는 근사 연산을 수행하면 연산 결과에 미치는 영향이 상대적으로 크므로 정확한 부분곱 생성기와 부분곱 누적기를 이용하여 연산을 수행한다. 따라서, 하위 40%, 37.5%, 30% 혹은 20% 이하의 비트들에 대하여 실시예에 의한 부분곱 생성기와 부분곱 누적기를 이용하여 연산을 수행하여도 연산 결과에 큰 영향을 미치지 않는다.
- [0029] 도 1로 예시된 본 실시예에 의한 곱셈기는 예시적으로 세 페이지로 동작한다. 제1 페이지(P1)에서, 부분곱 연산기는 승수와 피승수에 대하여 연산을 수행하여 복수의 부분곱들을 생성한다. 후술할 바와 같이 본 실시예에 의한 부분곱 연산기는 오차가 발생하며, 발생한 오차는 정확한 연산 결과에 비하여 큰 쪽으로 집중된다.
- [0030] 부분곱의 미리 정해진 하위 비트들에 대하여 청색 사각형으로 도시된 부분곱 누적기로 누적한다. 마찬가지로 후술할 바와 같이 부분곱 누적기는 오차가 발생하며, 발생한 오차는 정확한 연산 결과에 비하여 작은 쪽으로 집중된다. 나머지 상위 비트들은 근사 연산을 수행하면 연산 결과에 미치는 영향이 크므로 정확한 연산 결과를 얻을 수 있는 부분곱 누적기(검정색 사각형)로 연산을 수행한다.
- [0031] 도시된 실시예에서, 본 실시예에 의한 부분곱 누적기는 부분곱들에서 동일한 인덱스를 가지는 네 비트(혹은 세 비트)를 이용하여 누적 연산을 수행할 수 있다. 부분곱 누적 결과는 두 비트로, 입력된 비트에 포함된 “1”의 개수를 계수한 결과에 상응한다.
- [0032] 제2 페이지(P2)에서, 제1 페이지(P1)에서 연산된 결과를 다시 합산한다. 도시된 바와 같이 본 실시예에 의한 부분곱 누적기는 미리 정해진 하위 비트에 대하여 누적 연산을 수행한다. 제2 페이지에서는 2의 보수를 연산하기 위하여 추가된 “1” (검정 사각형) 또는 세 비트의 입력과 함께 추가되는 “0” 을 함께 포함하여 연산할 수 있다. 연산 결과는 각각 두 비트로 출력된다.
- [0033] 이어지는 제3 페이지(P3)에서는 전가산기(Full adder)를 이용하여 제2 페이지(P2)의 연산결과를 더하여 곱셈 결과를 형성한다.
- [0034] 상술한 바와 같이 본 실시예에 의한 부분곱 생성기는 LSB(least significant bit) 에 가까운 부분곱들을 연산한다. 또한, 본 실시예에 의한 부분곱 누적기는 LSB(least significant bit) 에 가까운 부분곱들을 합산한다. 후술할 바와 같이 본 실시예에 의한 부분곱 생성기는 에러 집중화된 부분곱을 생성하며, 부분곱들이 집중화된 에러는 정확한 부분곱 대비 큰 값을 가진다. 또한, 본 실시예에 의한 부분곱 누적기는 에러 집중화된 합을 생성하며, 부분곱들이 누적되어 형성되는 에러는 정확한 부분곱 대비 작은 값을 가진다. 따라서, 본 실시예에 의한 부분곱 생성기와 부분곱 누적기를 사용함으로써 오차를 평균화하여 제거(even-out)할 수 있다는 장점이 제공된다.
- [0035] 아래의 표 1은 본 실시예에 의한 부분곱 생성기의 동작을 설명하기 위한 표이다.

표 1

		$b_{2i+1}b_{2i}b_{2i-1}$							
		000	001	011	010	110	111	101	100
$a_j a_{j-1}$	00	0	0	0	0	1	0→1	1	1
	01	0	0	1→0	0	1	0→1	1	0→1
	11	0→1	1	1	1	0→1	0→1	0→1	0→1
	10	0→1	1	0→1	1	0→1	0→1	0→1	1

[0036]

[0037]

표 1에서 종래의 부스 인코더로 인코딩된 값을 검정색 값으로 표시하였으며, 본 실시예 의한 부분곱 생성기로 인코딩된 값을 적색과 청색으로 도시하였다. 청색으로 도시된 값은 종래의 부분곱 생성기인 부스 인코더에 비하여 인코딩된 값이 감소하는 것으로 인코딩되는 것을 나타내며, 적색으로 도시된 값은 종래의 부분곱 생성기인 부스 인코더에 비하여 인코딩된 값이 증가하는 것으로 인코딩되는 것을 나타낸다.

[0038]

또한, 표 1로 예시된 것과 같이 인코딩된 값이 감소하는 경우에 비하여 인코딩된 값이 증가하는 경우가 더 많다. 따라서, 본 실시예에 의한 부분곱 생성기는 부분곱을 생성할 때 예러가 증가하는 방향으로 집중화된 부분 곱을 형성하는 것을 확인할 수 있다.

[0039]

도 2(a)는 표 2로 예시된 종래 기술에 의한 부스 인코더의 게이트 레벨 회로도이고, 도 2(b)는 본 실시예에 의한 부분곱 생성기를 게이트 레벨로 도시한 회로도이다. 도 2(a)를 참조하면, 상기한 표 1에 따라 정확한 연산 결과를 얻을 수 있는 종래 기술의 부스 인코더는 4 개의 XOR 게이트, 하나의 2-입력 AND 게이트 하나의 3-입력 AND 게이트 하나의 OR 게이트 및 하나의 인버터를 포함한다.

[0040]

그러나, 이에 반하여 도 2(b)로 예시된 본 실시예에 의한 부분곱 생성기는 단일한 2 입력 OR 게이트로 형성할 수 있어 면적 측면에서 효율적이라는 장점이 제공되며, 나아가 소모 전력 측면에서 경제적이라는 장점이 제공된다.

[0041]

도 3(a)는 종래 기술에 의한 부스 인코더(booth encoder)로 십진수 13에 상응하는 이진수 01101과 2에 상응하는 이진수 10과의 연산을 예시하는 도면이다. 도 3(a)를 참조하면, 회색 박스로 도시된 최초 연산을 수행하는 승수(multiplier)의 비트들 $b_1b_0b_{-1}$ 은 100이다. 또한 최초 연산을 수행하는 피승수(multiplicand)의 비트들 a_0a_{-1} 은 10이다. 이들에 대한 연산 결과는 표 1에 도시된 것과 같이 1로 연산된다. 다음 연산의 대상인 피승수 비트들은 a_1a_0 이고, 01이므로 표 1로 도시된 것과 같이 연산 결과는 0이다. 이어서, 다음 연산의 대상인 피승수 비트들은 a_2a_1 은 10이므로 연산 결과는 1이다. 이와 같이 연산하면 부분곱은 1100101로 연산된다. 또한, 승수(multiplier) $b_1b_0b_{-1}$ 은 100으로 2의 보수(2's complement)를 형성하기 위하여 연산된 부분곱의 LSB에 1을 더한다.

[0042]

도 3(b)는 본 실시예에 의한 부분곱 생성기로 동일한 승수와 피승수를 대상으로 부분곱의 연산하는 과정을 예시하는 도면이다. 표 1과 도 3(b)를 참조하면, 본 실시예에 의한 부분곱 생성기로 하위 세 비트를 포함하여 부분 곱을 생성하는 것을 예시한다. 본 실시예에 따른 부분곱 연산은 도 3(b)에서 파선 상자로 도시되었다. 회색 박스로 도시된 연산 대상인 승수의 비트들 $b_1b_0b_{-1}$ 은 100이고 피승수 비트들은 a_0a_{-1} 은 10이므로 표 1에 도시된 것과 같이 1로 연산된다. 다음 연산 대상인 피승수 비트들은 a_1a_0 이고, 01이므로 연산 결과는 정확한 연산 결과와 달리 1이다. 이어서, 다음 연산 대상인 피승수 비트들 a_2a_1 은 10이므로 연산 결과는 1이다. 이어서, 하위 세 비트를 벗어나므로 정확한 연산을 수행하도록 한다. 피승수 a_3a_2 은 11이므로 연산 결과는 0이다. 이와 같이 연산하며, 부호 비트 확장(sign bit extension)하면 부분곱은 1100101로 연산된다. 또한, 승수(multiplier) $b_1b_0b_{-1}$ 은 100으로 2의 보수(2's complement)를 형성하기 위하여 LSB에 1을 더한다.

[0043]

도 3(a)와 도 3(b)로 예시된 전체 연산 결과를 대비한다. 도 3(a)로 예시된 정확한 연산 결과는 십진수 26에 상응하는 011010을 얻을 수 있다. 도 3(b)로 예시된 본 실시예에 의한 연산 결과는 십진수 28에 상응하는 011100

을 얻을 수 있다. 즉, 본 실시예에 따라 연산된 부분곱은 11100111이고, 정확하게 연산된 부분곱은 11100101이며, 두 부분곱은 서로 LSB 측 마지막 두 번째 비트만 상이한 것을 알 수 있다. 또한, 전체적으로 연산된 곱셈 결과는 28로 정확한 연산 결과인 26와 대비 에러가 작은 것을 확인할 수 있다.

[0044] 상술한 바와 같이 본 실시예에 의한 부분곱 생성기는 종래 기술에 의한 부스 인코더 대비 면적과 소모 전력에서 우수한 효율을 가지는 것을 확인할 수 있으며, 연산 결과의 측면에서도 높지 않은 에러를 가지는 것을 확인할 수 있다.

[0045] 나아가, 본 실시예에 의한 부분곱 생성기로부터 발생된 에러들은 대부분 정확한 연산 결과에 비하여 크게 집중되어 있다. 이러한 에러들은 후술할 바와 같이 부분곱 누적기를 통하여 오차를 평균화하여 제거(even-out)할 수 있다.

[0046] 이하에서는 표 2, 도 1, 도 4를 참조하여 본 실시예에 의한 에러가 집중화된 부분곱 누적기를 설명한다.

표 2

		$P_{4i+1,j}P_{4i,j}$			
		00	01	11	10
$P_{4i+3,j}P_{4i+2,j}$	00	00	01	10→01	01→00
	01	01→00	10→01	11→01	10→00
	11	10	11	11	11→10
	10	01→00	10→01	11→01	10→00

[0047]

[0048] 도 1에서 본 실시예에 의한 부분곱 생성기가 형성한 에러가 집중화된 부분곱은 적색점으로 도시되었으며, 본 실시예에 의한 에러가 집중화된 부분곱 누적기는 청색 사각형으로 도시되었다. 도 1 및 표 2로 예시된 바와 같이 본 실시예에 의한 에러가 집중화된 부분곱 누적기는 도 1에서 서로 다른 부분곱에 포함되나, 동일한 열 인덱스 (j)를 가져 동일한 컬럼에 위치하는 네 비트의 입력인 $P_{4i+1,j}$, $P_{4i,j}$, $P_{4i+3,j}$, $P_{4i+2,j}$ 을 제공받을 수 있다. 입력으로 제공되는 네 비트는 도 1에서 상기한 바와 같이 청색 사각형에 포함된 네 개의 적색점으로 도시되었다. 일 실시예에서, 후술할 바와 같이 네 비트의 입력 중 $P_{4i+1,j}$ 은 연산 결과에 영향을 미치지 않으므로 부분곱 누적기는 이를 입력받지 않을 수 있다.

[0049] 본 실시예에 의한 부분곱 누적기는 입력된 셋 이상의 비트에 포함된 “1”의 개수에 상응하는 값을 출력한다.

[0050] 표 2를 참조하면 입력된 비트들에 포함된 “1”의 정확한 개수는 검정색 글씨로 표시되었으나, 본 실시예에 의한 부분곱 누적기는 청색 글씨로 표시되었다. 표 2로 예시된 바와 같이 본 실시예의 부분곱 누적기는 정확한 계수 결과에 비하여 더 작은 값으로 계수하는 것을 확인할 수 있다.

[0051] 도 4(a)는 종래 기술에 의한 부분곱 누적기의 게이트 레벨 회로도이고, 도 4(b)는 본 실시예에 의한 부분곱 누적기의 게이트 레벨 회로도이다. 도 4(a)를 참조하면, 종래 기술에 의한 부분곱 누적기는, 네 개의 XOR 게이트, 네 개의 AND 게이트, 두 개의 OR 게이트 및 두 개의 인버터를 포함한다. 그러나, 본 실시예에 의한 부분곱 누적기는 하나의 선로와 하나의 2 입력 AND게이트로 구현할 수 있다. 즉, 본 실시예의 부분곱 누적기는 종래 기술 대비 우수한 면적 효율과 소모 전력 특성을 가지는 것을 알 수 있다.

[0052] 표 2를 참조하면, 본 실시예의 부분곱 누적기의 계수 결과는 정확한 계수 결과에 비하여 더 작은쪽으로 오차가 집중된 것이다. 이것은 정확한 연산 결과에 비하여 더 큰 쪽으로 오차가 집중된 본 실시예에 의한 부분곱 생성기와 함께본 실시예의 부분곱 누적기를 사용함으로써 부분곱 생성기에 의하여 생성된 오차와 부분곱 누적기가 생성한 오차를 평균화하여 제거(even-out)할 수 있다.

[0054] **모의 실험예**

[0055] 도 5(a)와 도 5(b)는 각각 본 실시예에 의한 부분 곱 생성기와 부분곱 누적기에서 근사 비트 수에 따른 오차 분

포를 도시한 도면이다. 도 5(a) 및 도 5(b)를 참조하면, 상술한 바와 같이 부분 곱 생성기에서 발생하는 오차는 정확한 값에서 큰 쪽으로 발생하고, 부분곱 누적기에서 발생하는 오차는 정확한 값에서 작은 쪽으로 발생하는 것을 확인할 수 있다. 나아가 부분 곱 생성기와 부분곱 누적기에서 발생하는 오차는 하위 12 비트까지 미미한 것을 확인할 수 있다.

[0056] 도 6은 본 실시예에 의한 부분 곱 생성기와 부분곱 누적기를 동시에 사용하여 곱셈기를 구성하였을 때의 최종 오차를 도시한 도면이다. 도 6을 참조하면, 하위 12 비트 까지 본 실시예에 의한 부분 곱 생성기와 부분곱 누적기를 이용하여 연산을 수행하여도 오차가 미미함을 확인할 수 있으며, 하위 14비트 까지도 허용할 수 있을 정도의 오차 수준을 보이는 것을 확인할 수 있다.

[0057] 도 7(a) 및 도 7(b)는 각각 FIR filtering과 CNN기반 이미지 분류에 본 실시예에 의한 부분 곱 생성기를 적용할 때의 정확도와 에너지 효율을 나타낸 도면이다. 도 7(a) 및 도 7(b)를 참조하면, 본 실시예에 의한 부분 곱 생성기를 이용하여 FIR Filtering과 CNN기반 이미지 분류를 수행할 때, 정확한 곱셈기 대비 각각 28%와 34% 적은 에너지로 곱셈을 수행할 수 있음을 확인할 수 있다.

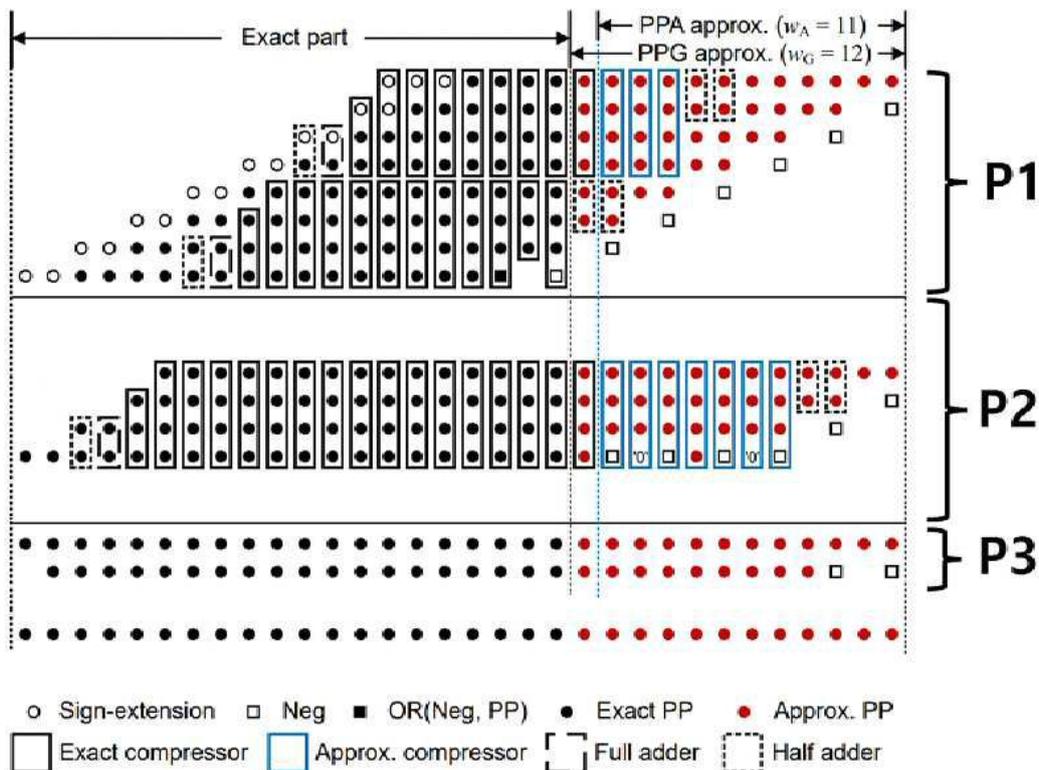
[0059] 본 발명에 대한 이해를 돕기 위하여 도면에 도시된 실시 예를 참고로 설명되었으나, 이는 실시를 위한 실시예로, 예시적인 것에 불과하며, 당해 분야에서 통상적 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시 예가 가능하다는 점을 이해할 것이다. 따라서, 본 발명의 진정한 기술적 보호범위는 첨부된 특허청구범위에 의해 정해져야 할 것이다.

부호의 설명

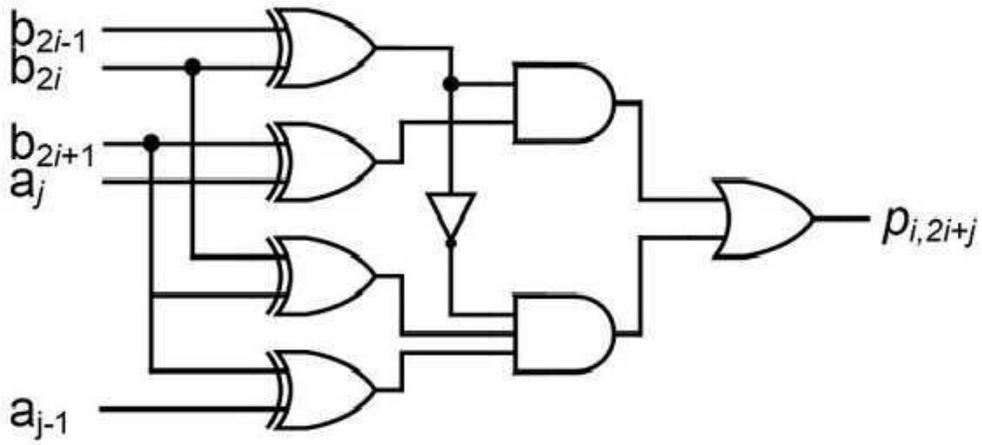
- [0060] 10: 부분곱 생성기
- 20: 부분곱 누적기

도면

도면1



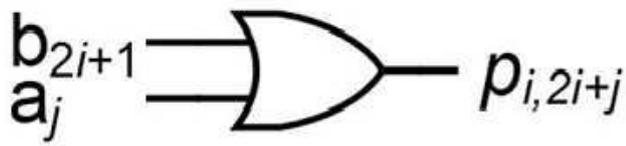
도면2



<종래 기술>

(a)

10



(b)

도면3

				...					
	0	0	0	1	1	0	1	0	
				0	0	1	0	0	
1	1	1	0	0	1	0	1		
0	0	1	1	0	1				
0	0	0	0				1		
0	0	0	1	1	0	1	0		

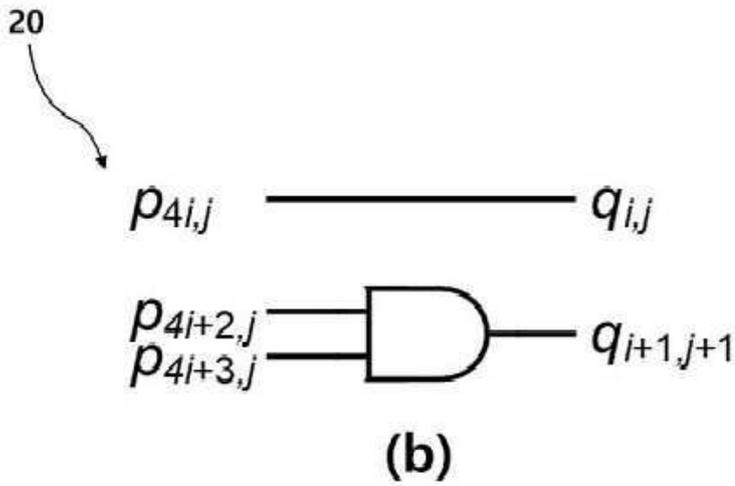
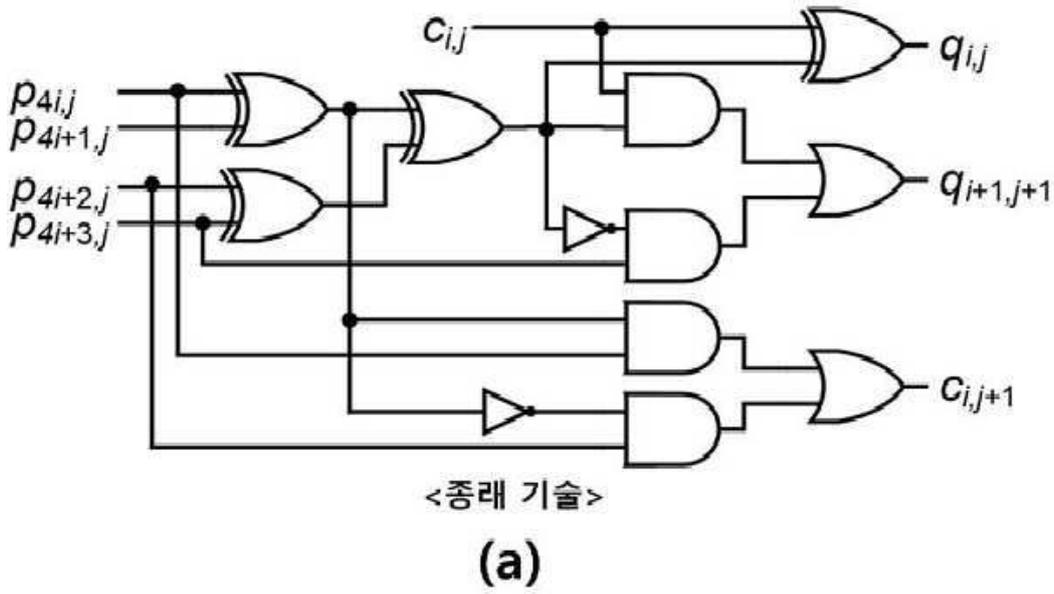
<종래 기술>

(a)

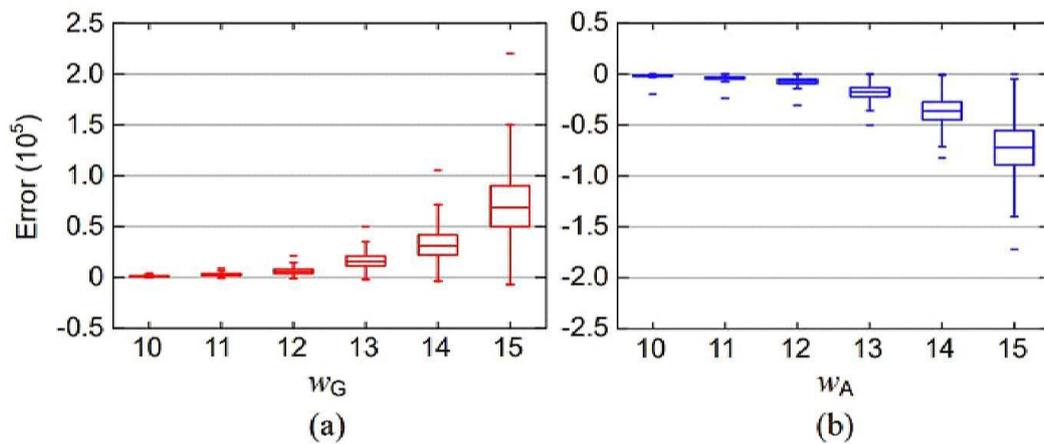
				...					
	0	0	0	1	1	0	1	0	
				0	0	1	0	0	
1	1	1	0	0	1	1	1		
0	0	1	1	0	1				
0	0	0	0				1		
0	0	0	1	1	1	0	0		

(b)

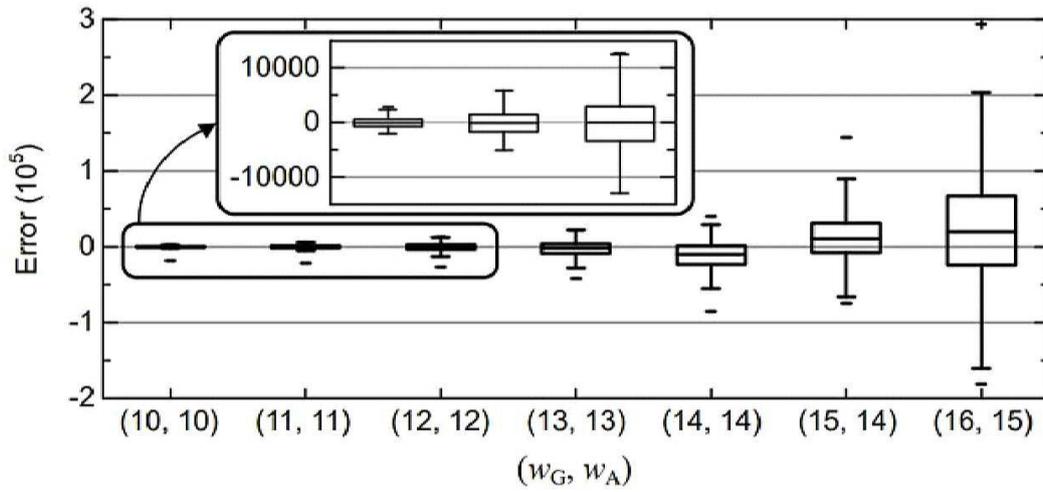
도면4



도면5



도면6



도면7

