

(52) CPC특허분류
G06N 20/00 (2021.08)

명세서

청구범위

청구항 1

추계학적 반올림 회로로서,

세 개 이상의 CSA 입력, CSA 합 출력 및 CSA 자리 올림 출력을 갖는 자리 올림 보존 가산기(CSA) 로직으로서, 상기 세 개 이상의 CSA 입력으로부터의 특정 CSA 입력에 난수 값이 제시되는, 상기 CSA 로직;

적어도 복수의 가산기 입력 및 합 출력을 갖는 가산기 로직을 포함하고;

상기 CSA 로직의 상기 CSA 자리 올림 출력은 상기 가산기 로직의 상기 복수의 가산기 입력의 제1 입력과 연결되고, 상기 CSA 로직의 상기 CSA 합 출력은 상기 가산기 로직의 상기 복수의 가산기 입력의 제2 입력과 연결되며;

상기 가산기 로직의 상기 합 출력의 최상위 비트들의 특정 수는 상기 CSA 로직의 상기 세 개 이상의 CSA 입력으로부터, 상기 특정 CSA 입력 이외의 적어도 두 CSA 입력의 추계학적으로 반올림된 합을 나타내는, 회로.

청구항 2

청구항 1에 있어서, 상기 세 개 이상의 CSA 입력은 누산된 입력 및 누산되지 않은 입력을 포함하고, 상기 누산된 입력은 상기 누산되지 않은 입력보다 더 많은 비트 폭을 갖는, 회로.

청구항 3

청구항 1에 있어서, 상기 세 개 이상의 CSA 입력은 누산된 입력 및 누산되지 않은 입력을 포함하고, 상기 누산되지 않은 입력 및 상기 누산된 입력은 상기 특정 CSA 입력보다 더 많은 비트 폭을 갖는, 회로.

청구항 4

청구항 1에 있어서,

CSA 자리 올림 레지스터;

CSA 합 레지스터를 더 포함하고;

상기 CSA 자리 올림 출력은 상기 CSA 자리 올림 출력이 상기 CSA 자리 올림 레지스터에 연결되고 상기 CSA 자리 올림 레지스터가 상기 제1 입력에 연결됨으로써 상기 가산기 로직의 상기 제1 입력에 연결되고;

상기 CSA 합 출력은 상기 CSA 합 출력이 상기 CSA 자리 올림 레지스터에 연결되고 상기 CSA 자리 올림 레지스터가 상기 제2 입력에 연결됨으로써 상기 가산기 로직의 상기 제2 입력에 연결되고;

상기 CSA 자리 올림 레지스터는 상기 세 개 이상의 CSA 입력의 제1 CSA 입력에 연결되고 상기 CSA 합 레지스터는 상기 세 개 이상의 CSA 입력의 제2 CSA 입력에 연결되며;

상기 세 개 이상의 CSA 입력의 제3 CSA 입력이 상기 특정 CSA 입력인, 회로.

청구항 5

청구항 1에 있어서, 윌리스 트리(Wallace Tree) 가산기 로직;

상기 윌리스 트리 가산기 로직의 특정 CSA 가산기 로직의 상기 특정 CSA 입력에 연결되는 다중화기 출력을 갖는 다중화기로서, 상기 다중화기의 적어도 하나의 입력은 난수이고 상기 다중화기의 적어도 다른 입력은 제로인, 상기 다중화기를 더 포함하고;

상기 특정 CSA 가산기 로직의 특정 CSA 자리 올림 출력은 상기 CSA 로직의 상기 CSA 자리 올림 출력이고 상기 가산기 로직의 상기 복수의 가산기 입력의 상기 제1 입력과 연결되며;

상기 특정 CSA 가산기 로직의 특정 CSA 합 출력은 상기 CSA 로직의 상기 CSA 합 출력이고 상기 가산기 로직의 상기 복수의 가산기 입력의 상기 제2 입력과 연결되는, 회로.

청구항 6

추계학적 반올림을 위한 방법으로서,

자리 올림 보존 가산기 로직에서, 복수의 입력 값을 수신하는 단계로서, 상기 복수의 입력 값의 적어도 하나의 입력 값은 무작위로 생성된 수인, 상기 복수의 입력 값을 수신하는 단계;

상기 무작위로 생성된 수에 적어도 부분적으로 기초하여, 자리 올림 출력 값 및 자리 올림 없는 합 출력 값을 생성하는 단계;

가산기 로직에서, 상기 자리 올림 출력 값 및 상기 자리 올림 없는 합 출력 값을 수신하는 단계;

상기 자리 올림 출력 값 및 상기 자리 올림 없는 합 출력 값에 적어도 부분적으로 기초하여, 상기 복수의 입력 값의 합을 생성하는 단계;

최상위 비트들의 특정 수를 상기 복수의 입력 값의 추계학적으로 반올림된 합으로서 선택하는 단계를 포함하는, 방법.

청구항 7

청구항 6에 있어서,

다수의 반복의 반복마다:

상기 자리 올림 보존 가산기에서, 상기 복수의 입력 값의 특정 입력 값을 수신하는 단계,

상기 특정 입력 값, 이전 자리 올림 없는 합 출력 값, 및 상기 복수의 반복의 이전 반복으로부터의 이전 자리 올림 출력 값에 기초하여, 중간 자리 올림 없는 합 출력 및 중간 자리 올림 출력을 생성하는 단계,

상기 다수의 반복의 다음 반복을 위해 상기 중간 자리 올림 없는 합 출력 및 상기 중간 자리 올림 출력을 저장하는 단계를 더 포함하는, 방법.

청구항 8

청구항 6에 있어서, 상기 자리 올림 보존 가산기 로직에서, 상기 복수의 입력 값을 수신하는 단계는:

다수의 반복의 각각의 반복마다 각각의 상기 복수의 입력 값을 수신하는 단계를 포함하며;

상기 다수의 반복의 마지막 반복에 수신된 상기 복수의 입력 값의 마지막 입력 값은 상기 무작위로 생성된 수인, 방법.

청구항 9

청구항 6에 있어서, 상기 무작위로 생성된 수를 배제한 상기 복수의 입력 값은 다른 자리 올림 보존 가산기 로직의 출력 값들인, 방법.

청구항 10

추계학적 반올림 방법으로서,

제1 가산기 로직에서, 복수의 입력 값의 중간 합을 생성하는 단계;

상기 중간 합의 최상위 비트에 기초하여, 상기 중간 합이 음인지 결정하는 단계;

무작위로 생성된 값을 음의 무작위로 생성된 값으로 변환하는 단계;

상기 중간 합 및 상기 무작위로 생성된 값에 기초하여, 제2 가산기 로직에서, 상기 복수의 입력 값의 합을 생성하는 단계;

상기 복수의 입력 값의 상기 합의 최상위 비트들의 특정 수를 상기 복수의 입력 값의 추계학적으로 반올림된 합으로서 선택하는 단계를 포함하는, 방법.

청구항 11

청구항 10에 있어서, 상기 무작위로 생성된 값을 상기 음의 무작위로 생성된 값으로 변환하는 단계는 적어도 상

기 무작위로 생성된 값의 최상위 비트를 셋팅함으로써 수행되는, 방법.

청구항 12

청구항 10에 있어서, 상기 무작위로 생성된 값을 상기 음의 무작위로 생성된 값으로 변환하는 단계는 적어도 상기 무작위로 생성된 값의 비트들을 부정함으로써 수행되는, 방법.

청구항 13

청구항 6 내지 12 중 어느 한 항에 나열된 방법을 수행하는 로직 회로.

청구항 14

삭제

청구항 15

삭제

청구항 16

삭제

청구항 17

삭제

청구항 18

삭제

청구항 19

삭제

청구항 20

삭제

청구항 21

삭제

청구항 22

삭제

발명의 설명

배경 기술

- [0001] 이 섹션에서 설명되는 접근법들은 추구될 수 있는 접근법들이고, 반드시 이전에 구상되거나 추구된 접근법들은 아니다. 따라서, 달리 명시되지 않는 한, 이 섹션에 설명되는 접근법들이 단순히 이 섹션에 포함된 것으로 인해 선행 기술로서의 자격이 있다고 가정되어서는 안 된다.
- [0002] 인공 지능의 성장과 함께, 기계 학습 기술들이 매우 다양한 응용 분야에 적용되고 있다. 기계 학습 모델 학습은 일반적으로 자원 집약적이고 그에 따라 일반적으로 수행을 위한 전용 컴퓨터 시스템들을 필요로 한다. 그러나, 기계 학습의 응용 분야들이 확장됨에 따라, 정확성을 희생하지 않고 공유 컴퓨팅 자원 환경에서 수행될 기계 학습 모델들의 학습에 대한 필요성이 증가하고 있다.
- [0003] 성능을 개선하기 위해, 정밀도가 감소된 수치 표현들이 기계 학습 모델들을 훈련하는데 사용될 수 있다. 예를 들어, 신경망들의 가중치들은 정밀도가 감소된 포맷을 가질 수 있고, 그에 따라 처리에 필요한 계산 자원들이

적다. 그러나, 일부 연산은 여전히 (일시적이지만) 정밀도가 보다 넓은 수치 표현들을 생성할 수 있다.

[0004] 정밀도가 보다 넓은 수치 표현들을 다시 정밀도가 감소된 것들로 감소시키는 한 가지 방법은 단순히 정밀도가 보다 넓은 수치 표현들을 잘라내는 것이다. 여분의 비트들을 자르는 것은 구현하기가 쉽지만(그리고 일반적으로 디폴트), 한 방향으로 값들(이들테면 가중치들)을 체계적으로 편향함으로써 혼란 오류들/낮은 정확도로 이어질 수 있다.

[0005] 정확도를 희생하지 않고 정밀도가 감소된 수치 표현들을 이용하기 위해, 평범한 잘라내기 대신 체계학적 반올림이 수행된다. 보다 넓은 정밀도의 수치 표현들의 체계학적 반올림은 편향 도입을 방지하고 그에 따라 결과적인 기계 학습 모델들의 정확도를 개선한다. 예를 들어, 보다 넓은 정밀도의 소수에 대한 체계학적 반올림은 보다 넓은 정밀도의 소수에서 떨어질 최하위 소수들에 비례하는 확률로 값을 반올림 또는 반내림된다. 따라서, 37.25의 값은 25% 확률로 38로 반올림되고 75% 확률로 37로 반내림된다.

[0006] 체계학적 반올림을 구현하기 위한 한 가지 접근법은 반올림의 결과를 내는 다수의 명령을 실행하는 것에 의한다. 소프트웨어 프로그램은 그러한 프로그램을 컴파일하는 동안, 하드웨어 프로세서에 의해 실행될 다수의 명령을 낼, 체계학적 반올림에 적절한 명령(들)을 포함할 수 있다. 다수의 명령을 처리할 때 높은 오버 헤드: 특히 다중 프로세서 주기, 잠재적인 다중 메모리 조회 및 파이프라인 중단 등이 발생한다.

도면의 간단한 설명

[0007] 도 1은 일 실시 예에서, 체계학적 반올림을 이용한 합산을 위한 예시적인 회로를 도시한 블록도이다.

도 2는 일 실시 예에서, 자리 올림 보존 가산기(CSA, carry-save adder) 로직 구성요소들의 예시적인 회로들을 도시한 블록도이다.

도 3은 일 실시 예에서, 입력 수들의 스트림을 누산하고 그러한 누산을 체계학적으로 반올림하기 위한 회로를 도시한 블록도이다.

도 4는 일 실시 예에서, 입력 수들의 다수의 스트림을 합산하고 그러한 합산을 체계학적으로 반올림하기 위한 회로를 도시한 블록도이다.

도 5는 일 실시 예에서, 합산을 그것의 부호에 기초하여 체계학적으로 반올림하는 회로를 도시한 블록도이다.

도 6은 일 실시 예에서, 부호 있는 합산의 체계학적 반올림을 위한 예시적인 회로를 도시한 블록도이다.

도 7은 일 실시 예에서, 두 개 이상의 수의 합을 체계학적으로 반올림하기 위한 프로세스를 도시한 흐름도이다.

도 8은 일 실시 예에서, 두 개 이상의 수의 부호 있는 합을 체계학적으로 반올림하기 위한 프로세스를 도시한 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0008] 다음 내용에서, 설명을 위해, 본 발명에 대한 완전한 이해를 제공하기 위해 많은 구체적인 세부 사항이 제시된다. 그러나, 본 발명이 이러한 구체적인 세부 사항들 없이도 실시될 수 있다는 것이 이해될 것이다. 다른 사례들에서, 주지된 구조들 및 디바이스들은 본 발명을 불필요하게 모호하게 하는 것을 피하기 위해 블록도 형태로 도시된다.

[0009] 개관

[0010] 보다 큰 코드 밀도, 보다 높은 성능, 보다 높은 프로세서 이용률 및 보다 낮은 전력을 달성하기 위해, 여기에 체계학적 반올림을 수행하기 위한 다양한 회로 및 기술이 설명된다. 일 실시 예에서, 난수가 생성되고 두 개 이상의 수의 합산에 가산되어 두 개 이상의 수의 체계학적으로 반올림된 합을 생성한다.

[0011] 예를 들어, 두 개 이상의 수의 샘플 합이 37.25라고 가정한다. 그러한 합을 체계학적으로 반올림할 때, 결과는 75% 확률로 37이고 25% 확률로 38이다. 0 내지 1 범위(1은 포함되지 않음)에서 난수가 생성될 때, 난수는 0 내지 0.75 범위(0.75는 포함되지 않음)에 있을 확률이 75%이고 0.75 내지 1 범위(1은 포함되지 않음)에 있을 확률이 25%이다. 그에 따라, 그러한 난수를 37.25의 합에 가산하는 것은 그 때의 75%, 37로 시작되는 결과(결과는 37.25 내지 38.0 범위(38.0은 포함되지 않음)에 있음), 및 그 때의 25%, 38로 시작되는 결과(결과는 38.0 내지 38.25 범위(38.25는 포함되지 않음)에 있음)를 낸다.

- [0012] 따라서, 추계학적으로 반올림된 보다 넓은 정밀도의 합이 보다 낮은 정밀도의 합을 계산하기 위해, 보다 넓은 정밀도의 합을 낼 수 있는 가수들을 합산하는 것 외에도, 일 실시 예에서, 난수가 생성되어 가산된다. 난수는 다른 가수들과 동시에 가산된다. 무작위로 생성된 수는 원하는 보다 낮은 정밀도 폭과 보다 높은 정밀도의 폭 사이의 차이와 동일한 폭을 갖는다. 추계학적으로 반올림된 합은 최상위 유효 숫자들의 보다 낮은 정밀도의 폭을 검색하고 나머지 결과를 버림으로써 생성된다. 예를 들어, 보다 넓은 폭의 합이 32 비트인 한편 추계학적으로 반올림된 보다 낮은 폭의 합이 8 비트인 경우, 합산에 대한 추가 가수로서 32-8=24 비트의 난수가 생성된다. 합산 이후, 32 비트 합의 최상위 8 비트는 추계학적으로 반올림된 8-비트 폭 합으로서 판독된다.
- [0013] 일 실시 예에서, 입력 값들의 추계학적으로 반올림된 합을 생성하기 위해 제시된 입력 값들에 난수 값을 가산하는데 자리 올림 보존 가산기(CSA) 로직이 사용된다. "자리 올림 보존 가산기(CSA, carry-save adder) 로직"이라는 용어는 세 개 이상의 입력을 갖고 자리 올림 출력(carry output) 및 자리 올림 없는 부분 합 출력(carry-less partial sum output)을 생성하는 회로를 지칭한다. 자리 올림 없는 부분 합 출력은 대응하는 숫자들을 가산하는 동안 생성된 자리 올림 값들을 고려하지 않은 모든 입력의 합이다. 대응하는 숫자들에 대한 미세상 자리 올림 값들은 CSA 로직의 자리 올림 출력으로 집계된다. 자리 올림 보존 가산기는 CSA 로직의 일례이다. 여기에 설명된 회로들 및 기술들은 설명을 위해 이진 자리 올림 보존 가산기를 지칭할 수 있지만, 임의의 다른 CSA 로직이 효과적으로 대체될 수 있다. 그에 따라, 그러한 회로들 및 기술들은 자리 올림 보존 가산기들로 제한되는 것으로 해석되어서는 안 되고, 임의의 CSA 로직에 의해 구현될 수 있다.
- [0014] CSA 로직은 가산기 로직에 연결되어 입력 값들의 총 합을 생성한다. "가산기 로직(adder logic)"이라는 용어는 두 개 이상의 입력을 갖고 두 개 이상의 입력에 제시되는 값들(임의의 생성된 자리 올림들을 포함)의 총 합을 생성하는 회로를 지칭한다. 여기에 설명된 회로들 및 기술들은 설명을 위해 자리 올림 완료 가산기를 지칭할 수 있지만, 가산기 로직으로서, 그러한 회로들 및 기술들은 전 가산기들에 제한되는 것이 아니라 임의의 가산기 로직으로 해석되어야 한다. 전 가산기의 비제한적인 예들은 리플 자리 올림 가산기 및 자리 올림 예측 가산기이다.
- [0015] 자리 올림 출력 및 부분 합 출력은 가산기 로직에 대한 입력들로서 제공된다. 가산기 로직의 최상위 비트들의 수는 CSA 로직의 둘 이상의 입력의 추계학적으로 반올림된 합을 나타낸다.
- [0016] 도 1은 일 실시 예에서, 추계학적 반올림을 이용한 합산을 위한 예시적인 회로를 도시한 블록도이다. 예시적인 회로는 누산기 레지스터(accumulator register)(105), CSA(110) 및 전 가산기(full adder)(120)를 포함한다. CSA(110)는 세 개의 입력(X31-X0, Y15-Y0 및 Z15-Z0)을 갖는다.
- [0017] 그러한 입력들 각각에 제시되는 이진 수들이 CSA(110) 및 전 가산기(120)에 의해 합산된다. 각각, CSA(110)의 자리 올림 출력 C31-0는 전 가산기(120)의 입력들 U31-U0와 연결되고, CSA(110)의 부분 합 출력 S31-S0는 전 가산기(120)의 입력들 V31-V0와 연결된다.
- [0018] CSA(110)는 입력들(X31-X0, Y15-Y0 및 Z15-Z0)의 대응하는 비트들의 부분 합산(자리 올림 없는 합산)을 수행하도록 구성된다. 부분 합은 출력 S31-S0에 제공된다. 각 대응하는 입력 비트들의 가산에 대한 자리 올림 수들의 집계는 출력 C31-C0에 제공된다. 전 가산기(120)는 입력들(U31-U0 및 V31-V0)에 제시되는 이진 값들의 자리 올림 완료 가산을 수행한다. 가산기(120)에 의해 생성된 총 합은 출력 31-0에 제공된다.
- [0019] 이러한 예에서, CSA 입력들 X31-X0는 누산기 레지스터(105)에 연결된다. 그에 따라, 누산기 레지스터(105)에 저장된 수는 CSA(110)에 대한 입력으로서 입력들 X31-X0에 제시된다. 누산된 수는 CSA(110) 입력들 Y15-Y0에 제시되는 16-비트 입력 수에 가산된다. CSA(110)에 대한 다른 입력은 16-비트 난수이며, 이는 CSA(110) 입력들 Z15-Z0에 제시된다.
- [0020] 이러한 예에서 난수 입력은 16-비트 입력이므로, 전 가산기(120)의 하위 16 비트인 출력 15-0은 폐기된다. 나머지 비트들의 출력인 출력 31-16은 입력 수와 레지스터(105)에 저장된 수의 추계학적으로 반올림된 합이다.
- [0021] 난수 레지스터
- [0022] 일 실시 예에서, 난수는 선형 귀환 이동 레지스터(linear-feedback shift register)에 의해 생성된다. 그러한 실시 예에서, 선형 귀환 이동 레지스터는 CSA 로직의 난수 입력에 연결된다. 선형 귀환 이동 레지스터는 임의의 비트 길이일 수 있고 그러한 비트 길이의 의사 난수를 생성한다.
- [0023] 다른 실시 예에서, 레지스터는 CSA 로직의 난수 입력에 연결된다. 난수는 합의 추계학적 반올림을 계산하기 전에, 생성되어 레지스터에 저장된다.

- [0024] 추계학적 반올림 회로에 대한 최적화
- [0025] 일 실시 예에서, CSA 로직은 난수 입력이 제시되지 않는 하나 이상의 입력 비트에 대해 최적화된다. 그러한 실시 예에서, CSA 로직 입력에 제시되는 난수는 CSA 로직에 대한 적어도 하나의 다른 입력보다 적은 비트 폭을 갖는다. 난수 입력을 갖지 않는(또는 그 외 난수 입력에 대해 항상 제로 값으로 셋팅되는) CSA 로직의 회로 구성 요소들은 제거되거나 최적화될 수 있다. 그렇게 하면 회로의 밀도, 전력 소비 및 성능이 개선된다.
- [0026] 일 실시 예에서, CSA 로직의 입력은 누산된 입력이다. 누산된 입력은 누산된 입력에, 이전 연산의 중간 결과가 제시되기 때문에 CSA 로직의 다른 입력들보다 넓다. 이전 연산으로 인해 보다 넓은 폭의 누산된 입력에 의해 사용되는 추가 비트(들)가 생성되었을 수 있다. 예를 들어, 누산된 입력은 폭이 보다 넓고 이전 연산들의 중간 결과가 저장되는 누산기 레지스터에 연결될 수 있다.
- [0027] 그에 따라, CSA 로직의 누산되지 않은 입력은 적어도 CSA 로직의 누산된 로직보다 적은 비트 폭을 갖는다. 일 실시 예에서, 누산되지 않은 수 입력을 갖지 않는(또는 그 외 누산되지 않은 입력에 대해 항상 제로 값으로 셋팅되는) CSA 로직의 회로 구성요소들은 난수 입력이 없는 구성요소들과 유사하게 제거되거나 최적화된다.
- [0028] 도 2는 일 실시 예에서, CSA 로직 구성요소들의 예시적인 회로들을 도시한 블록도이다. 이러한 예에서, CSA 로직의 누산된 입력은 32-비트 입력이고, 누산되지 않은 입력은 16-비트 입력이며, 난수 입력은 8-비트 입력이다. 회로(200)는 자리 올림 출력의 최하위 비트 C0와 부분 합 출력의 최하위 비트 S0를 내는 샘플 로직 회로이다. 출력은 입력에 누산된 입력의 최하위 비트 A0, 입력에 누산되지 않은 입력의 최하위 비트 B0, 및 입력에 난수 입력의 최하위 비트 C0를 제시함으로써 생성된다.
- [0029] 회로(220)는 자리 올림 출력의 제8 비트 C8과 부분 합 출력의 제8 비트S8을 내는 샘플 로직이다. 최하위 비트들에 대해 도시된 것으로서 회로 로직, 회로(200)를 사용하는 대신, 회로(220)가 최적화된다. 최적화는 난수 입력(C)이 없는 것에 기인하는데, 이는 난수 입력이 8-비트의 보다 적은 폭의 입력이고 그에 따라 비트 0에서 7까지만 의미있는 입력들을 갖기 때문이다. 난수 입력이 없기 때문에, 회로(200)의 204, 208 및 210의 동등한 로직 게이트들이 회로(220)에서 제거된다. 회로(200)의 로직 게이트(202)와 동등한 로직 게이트(222), 및 회로(200)의 로직 게이트(208)와 동등한 로직 게이트(228)가 제8-비트 부분 합 S8 및 제8-비트 자리 올림 출력 C8을 생성하는데 사용된다.
- [0030] 회로(230)는 샘플 CSA 로직의 제16-비트 입력에서 시작되는 난수 입력 및 누산되지 않은 수 입력 양자가 없는 것에 기초한 회로(220)의 추가 최적화이다. 추가 최적화를 통해, 비트들 16 내지 31에 대한 자리 올림 출력 및 부분 합 출력을 생성하는데 어떠한 로직 게이트도 사용되지 않는다.
- [0031] 입력 수 스트림(들) 누산
- [0032] 일 실시 예에서, CSA 로직의 부분 합 출력 및 CSA 로직의 자리 올림 출력은 각각의 중간 레지스터들에 연결된다. 중간 레지스터들은 그 자체로 전 가산기의 입력에 그리고 귀환 경로로 CSA 로직의 적어도 두 개 입력에 연결된다. CSA 로직의 나머지 입력(들)은 합산될 각각의 입력 수 스트림(들)으로 제시된다. 각 사이클마다, 입력 수는 이전 사이클의 부분 합과 부분적으로 합산되고, 자리 올림 출력은 중간 레지스터들에 저장되며, 새로운 부분 합 및 새로운 자리 올림 출력이 중간 레지스터들에 저장된다.
- [0033] 일 실시 예에서, 입력 스트림의 마지막 입력 수가 제시된 후, 다음 사이클에서, CSA 로직의 입력 스트림 입력에 난수가 제시된다. 난수는 난수를 포함하는 레지스터 또는 선형 귀환 이동 레지스터로부터의 출력을 다중화하는 것에 기초하여 제시된다. 입력 스트림 입력에 난수를 제시함으로써, 최종 누산된 자리 올림 및 부분 합이 난수와 함께 누산된다. 그 결과로 초래된 누산의 최상위 비트들의 수는 제시된 입력 스트림의 추계학적으로 반올림된 합이다.
- [0034] 다른 실시 예에서, 중간 레지스터들 중 하나는 난수로 초기화된다. 그에 따라, 첫 번째 누산 사이클에서, 난수가 입력 스트림의 첫 번째 수와 누산된다. 그에 따라, 입력 스트림을 처리하는 모든 사이클 이후에, 출력의 최상위 비트들이 입력 스트림의 추계학적으로 반올림된 합을 나타낸다.
- [0035] 도 3은 일 실시 예에서, 입력 수들의 스트림을 누산하고 그러한 누산을 추계학적으로 반올림하기 위한 회로를 도시한 블록도이다. 예시적인 회로는 누산기 레지스터들(315 및 325), CSA(310) 및 전 가산기(320)를 포함한다. CSA(310)는 두 개의 32-비트 입력(X31-X0 및 Y31-Y0), 및 하나의 16-비트 입력(Z15-Z0)을 갖는다.
- [0036] 그러한 입력들 각각에 제시되는 이전 수들은 CSA(310) 및 전 가산기(320)에 의해 합산된다. CSA(310)의 자리 올림 출력 C31-C0는 레지스터(315)에 연결되며, 이는 그 자체로 전 가산기(320)의 입력 U31-U0와 연결된다.

CSA(110)의 부분 합 출력 S31-S0는 레지스터(325)와 연결되며, 이는 그 자체로 전 가산기(320)의 입력 V31-V0와 각각 연결된다.

[0037] 레지스터들(315 및 325)은 이러한 예의 귀환 루프에서, 각각 입력(X31-X0 및 Y31-Y0)에 각각 연결된다. 레지스터들(315 및 325)은 제로로 초기화된다.

[0038] 각 사이클마다, 16-비트 입력 수가 다중화기(330)에 의해 다중화되어 입력들(Z15-Z0)에 제시된다. CSA(310)는 입력 수를 평가하고 레지스터들(315 및 325)로부터 제시된 이전에 평가된 자리 올림 출력 및 부분 합과 합산한다. 예를 들어, 첫 번째 사이클에서, 레지스터(315 및 325)는 제로를 포함함에 따라, CSA(310)에 의한 평가 결과 입력 수를 레지스터(325)에 제로 값들과 부분 합 출력으로서 저장한다. 입력 수에 제로를 가산하면 자리 올림 출력을 내지 않으므로 레지스터(315)는 계속 제로를 저장한다. 다음 사이클에서는, CSA(310)와 레지스터(325)의 귀환 연결로 인해 입력 Y31-0에 이전 입력이 다시 제시된다. 이러한 사이클에서, 새로 제시된 입력 수는 이전 입력 수와 누산되어 새로운 부분 합을 내 레지스터(325)에 저장되고 합의 자리 올림 출력을 내 레지스터(315)에 저장된다. 사이클은 입력 스트림의 모든 입력 수가 CSA(310)의 Z15-0 입력에 제시될 때까지 반복된다.

[0039] 스트림의 마지막 입력 수를 제시하는 마지막 사이클 이후의 사이클에서, 16-비트 다중화기(330)에 대한 제어 신호는 CSA(310)의 Z15-0 입력에 제시될 난수 입력을 선택한다. 난수는 CSA(310)에서 각각, 입력들(X31-0 및 Y31-0)에 레지스터들(315 및 325)로부터 제시되는 누산된 자리 올림 값 및 누산된 부분 합 값과 부분적으로 합산된다. 각각의 레지스터들(315 및 325)에서의 결과 부분 합 및 결과 자리 올림 출력은 난수를 포함하고 전 가산기(320)에서 가산된다. 전 가산기(320) 출력의 최상위 비트들 31-16은 입력 스트림의 추계학적으로 반올림된 합을 나타내는 한편, 출력 비트들의 나머지는 폐기된다.

[0040] 일 실시 예에서, 회로는 세 개 이상의 입력 수의 합을 동시에 합산하고 추계학적으로 반올림한다. 회로는 윌리스 트리(Wallace tree) 가산기 로직을 포함하는 CSA 로직을 사용한다. 여기서 "윌리스 트리 가산기 로직"이라는 용어는 네 개 이상의 입력을 갖고 입력들에 제시되는 입력 수들의 자리 올림 출력 및 자리 올림 없는 부분 합 출력을 생성하는 회로를 지칭한다. 윌리스 트리 가산기는 윌리스 트리 가산기 로직의 일례이다. 여기에 설명된 회로들 및 기술들은 설명을 위해 윌리스 트리 가산기를 지칭할 수 있지만, 임의의 다른 윌리스 트리 가산기 로직이 효과적으로 대체될 수 있다. 그에 따라, 그러한 회로들 및 기술들은 윌리스 트리 가산기로 제한되는 것으로 해석되어서는 안 되고, 어느 윌리스 트리 가산기 로직에든 적용 가능하다.

[0041] 도 4는 일 실시 예에서, 입력 수들의 다수의 스트림을 합산하고 그러한 합산을 추계학적으로 반올림하기 위한 회로를 도시한 블록도이다. 이러한 예에서, 회로는 윌리스 트리 가산기를 사용한다. 윌리스 트리 가산기는 윌리스 트리 가산기의 입력들에 제시되는 다수의 입력 수를 동시에 누산하여 누산된 자리 올림 출력 및 누산된 부분 합 출력을 내는 연결된 CSA 가산기들의 네트워크를 포함한다.

[0042] 윌리스 트리 가산기(410)는 10개의 입력을 가지며, 입력들 중 두 개의 입력(C 및 S)이 도 3과 유사하게, 자리 올림 출력 누산기 레지스터(415) 및 부분 합 출력 누산기 레지스터(425)에 연결된다. 윌리스 트리 가산기의 각 CSA는 입력 수를 하나 더 적은 출력 수로 줄인다. 그 다음 그러한 CSA의 출력은 다른 CSA의 입력과 연결되어 자리 올림 출력(P) 및 부분 합 출력(Q)의 단지 두 개의 출력만 있을 때까지, 출력 수를 더욱 줄인다.

[0043] 윌리스 트리 가산기(410)의 출력들(P 및 Q)은 CSA(413)의 입력들에 연결된다. CSA(413)의 세 번째 입력은 다중화기(430)에 연결된다. 다중화기(430)의 제어 입력(C1)에 기초하여, 난수 입력 또는 제로 값 중 어느 하나가 CSA(413)의 입력에 제시될 것으로 선택된다. 제로 값을 제시하면 CSA(413)의 출력 레지스터들(415 및 425)에서 X_1-X_8 의 입력들의 수 스트림들이 누적될 수 있다. 스트림들의 마지막 엔트리들에서, 다중화기(430)는 일 실시 예에서, 최종 누산에 가산될 난수에 대한 난수 입력을 선택한다. 다른 실시 예들에서, 난수 입력은 임의의 다른 누산 사이클에서 다중화기(430)에 의해 선택된다.

[0044] CSA(413)의 출력들은 자리 올림 출력 레지스터(415) 및 부분 합 출력 레지스터(425)에 연결되며, 이들은 그 자체로 전 가산기(420)에 연결된다. 전 가산기(420)는 CSA(413)의 자리 올림 출력 및 부분 합의 전 가산을 출력 레지스터들(415 및 425)에 누산되는 것으로 평가한다. 그에 따라, 전 가산기(420)는 다수의 수 스트림의 합산들의 자리 올림 완료 결과들을 낸다. 마지막 사이클에서, 전 가산기(420) 출력의 최상위 비트들의 수는 윌리스 트리 가산기(410)의 입력들에 제시되는 수들의 입력 스트림의 추계학적으로 반올림된 합이다.

[0045] 음의 합 반올림

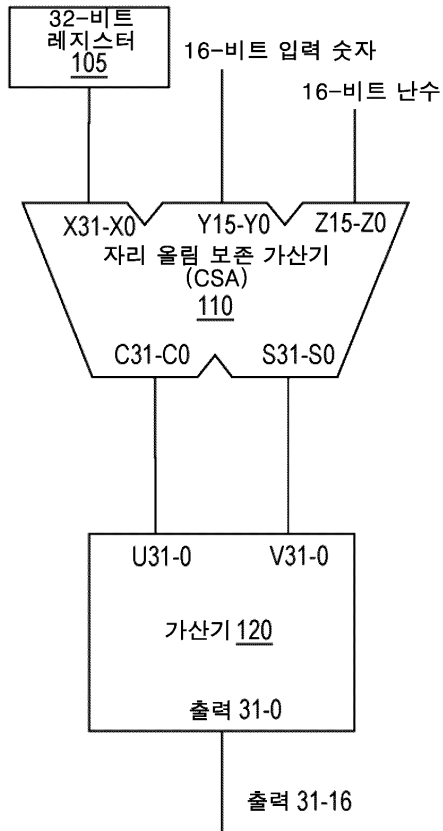
- [0046] 음의 합을 추계학적으로 반올림하기 위해 난수가 감소된다(또는 달리 언급하면, 음의 난수가 가산된다). 예를 들어, 두 개 이상의 수의 샘플 합이 -37.25이다. 그러한 음의 합을 추계학적으로 반올림할 때, 결과는 75% 확률로 -37이고 25% 확률로 -38이다. -1 내지 0 범위(-1은 포함되지 않음)에서 난수가 생성될 때, 난수는 -0.75 내지 0 범위(-0.75는 포함되지 않음)에 있을 확률이 75%이고 -1 내지 -0.75 범위(-1은 포함되지 않음)에 있을 확률이 25%이다. 그에 따라, 그러한 음의 난수를 -37.25의 합에 가산하는 것은 그 때의 75%, -37로 시작되는 결과(결과는 -38 내지 -37.25 범위(-38.0은 포함되지 않음)에 있음), 및 그 때의 25%, -38로 시작하는 결과(결과는 -38.25 내지 -38.0 범위에 있음)를 낸다.
- [0047] 일 실시 예에서, 음의 난수를 생성하기 위해서는, 양의 난수가 생성되고(예를 들어, 여기에 설명된 기술들을 사용하여) 그 다음 음수로 변환된다. 일 실시 예에서, 양의 난수는 양수의 각 비트를 부정함으로써 동일한 폭의 음수로 변환되어 1의 보수 음수를 낸다. 다른 실시 예에서는, 양의 난수의 각 비트가 부정된 다음 1의 값이 가산되어 양의 난수가 난수의 2의 보수로 변환된다. 또 다른 실시 예에서는, 무작위로 생성된 양수의 최상위 비트가 부정되어 양의 난수와 절대 값이 동일하지 않지만 음의 난수를 낸다.
- [0048] 일 실시 예에서, 입력 수들의 합을 추계학적으로 반올림하기 위해, 합은 추계학적 반올림 없이 생성된 다음 합의 부호(양수 또는 음수)에 기초하여 난수가 음의 난수로의 변환 여부에 관계없이 가산된다.
- [0049] 도 5는 일 실시 예에서, 합산을 그것의 부호에 기초하여 추계학적으로 반올림하는 회로를 도시한 블록도이다. 도 5의 샘플 회로는 전 가산기(540) 및 16-비트 XOR 게이트(550)를 포함한다. 32-비트 합은 입력 합 추계학적 반올림을 생성하기 위한 전 가산기(540)에 대한 입력 U31-0이다. 전 가산기(540)는 또한 들어오는 자리 올림 입력을 위한 입력 CIN을 전 가산기에 드러낸다. 입력 합 최상위 비트 U31은 또한 XOR 게이트(550)의 16-비트 입력의 각 비트에 연결된다. XOR 게이트의 다른 16-비트 입력은 양의 난수 입력에 연결된다. XOR 게이트(550)의 출력은 전 가산기(540)의 입력들 V15-V0에 연결된다.
- [0050] 그에 따라, 입력 합 최상위 비트 U31이 하이(합이 음수임을 나타냄)일 때, 각 난수 비트들이 값이 1과 XOR된다. 그러한 XOR 연산은 양의 난수에 대한 1의 보수를 낸다. 그 다음 1의 보수 난수가 입력들 V15-V0에 전 가산기(540)에 대한 입력으로서 제시되어 입력 U31-U0의 합과 가산된다. 추가적으로, 합 최상위 비트인 U31이 전 가산기(540)의 입력 CIN에 연결됨에 따라, 합과 음수 난수의 합산에 1의 값이 가산된다. 그러한 가산은 효과적으로 음의 난수를 제시된 양의 난수의 2의 보수로 만든다. 출력 31-16과 같은 전 가산기(540)의 출력으로부터의 결과의 다수의 최상위 비트가 추계학적으로 반올림된 음의 합으로서 선택된다.
- [0051] 입력 합이 양수인 경우에는, XOR 게이트(550)의 16-비트 입력이 제로 값을 갖는다. 그에 따라, XOR 게이트(550)에서의 XOR 연산은 난수 입력에 제시된 것과 동일한 값을 낸다. 유사하게, 입력 CIN은 제로 값을 가지므로, 전 가산기(540)에서 추가 값이 가산되지 않는다. 그에 따라, 전 가산기(540)의 출력은 입력 합과 변경되지 않은 양의 난수의 합산이다.
- [0052] 일 실시 예에서, 합 추계학적 반올림을 위한 회로는 합들의 추계학적 반올림을 위한 두 개의 서브-회로를 포함하며, 두 개의 서브-회로는 난수 입력의 부호가 상이하다. 하나의 서브-회로는 난수 입력에 양의 난수를 갖고, 다른 하나의 서브-회로는 난수 입력에 음의 난수를 갖는다. 두 개의 서브-회로를 사용하여 추계학적 반올림 합을 계산된 후, 두 추계학적 반올림 합 결과 중 어느 결과를 회로의 출력 결과로서 선택하는데 각각의 결과들의 최상위 비트들 중 하나 또는 양자가 사용된다. 예를 들어, 양의 난수 서브-회로의 결과의 최상위 비트가 양수이면, 결과가 회로의 출력으로서 선택된다. 양의 난수 서브-회로의 결과가 음수이면, 음의 난수 서브-회로의 결과가 사용된다.
- [0053] 도 6은 일 실시 예에서, 부호 있는 합산의 추계학적 반올림을 위한 예시적인 회로를 도시한 블록도이다. 예시적인 회로는 서브-회로들(610A 및 610B)을 포함한다. 서브-회로들(610A 및 B)은 입력 수들을 합산하고 추계학적으로 반올림된 수를 나타내는 동일한 회로를 갖는다. 서브-회로들(610A 및 B)은 여기서 논의된 회로들 중 어느 하나의 회로를 포함할 수 있다. 서브-회로들(610A 및 B)에 대한 입력으로서 제시되는 입력 수들은 동일하다. 그러나, 난수 입력은 인버터(640)에 의해 서브-회로 B에 대해 반전된다. 인버터(640)는 난수 입력의 최상위 비트에만 연결된다. 그러한 반전(단지 완료 또는 최상위 비트(MSB))은 난수 입력에 제시되는 양의 난수를 여전히 난수이지만 상이한 음수로 변환한다.
- [0054] 서브-회로들(610A 및 610B)의 출력들의 부호들(각각의 출력들의 최상위 비트들(MSB들)은 올바른 추계학적 반올림을 반영하는 서브-회로의 출력을 결정한다. 예를 들어, 다중화기(630)가 서브-회로(610B)의 최상위 비트들에 연결된다. 다중화기(630)는 서브-회로들(610B)의 부호가 음일 때 서브-회로(610A)의 출력을 선택하고 서브-회로

(610B)의 부호가 양일 때 서브-회로(610B)의 출력을 선택한다.

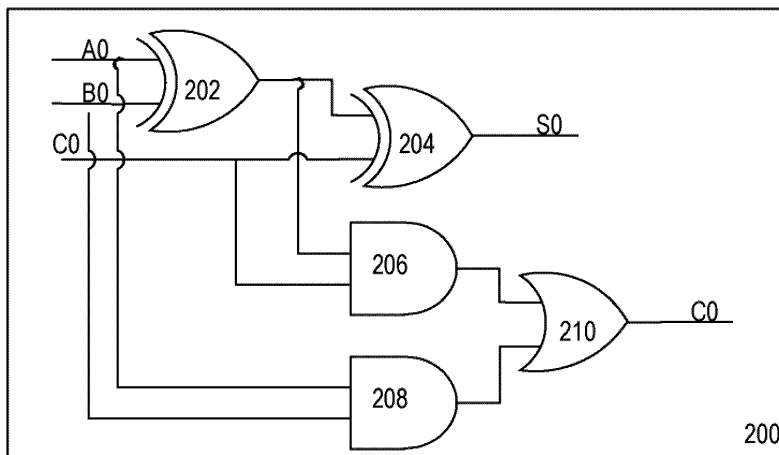
- [0055] 기능적 개요
- [0056] 도 7은 일 실시 예에서, 두 개 이상의 수의 합을 추계학적으로 반올림하기 위한 프로세스를 도시한 흐름도이다. 단계 705에서, 합산할 두 개 이상의 입력 수가 수신된다. 단계 710에서, 추계학적 반올림을 위해 난수가 수신된다. 도 7에 도시된 바와 같은 프로세스가 반복적으로 수행될 수 있지만, 난수는 프로세스의 매 반복보다는 단일 반복 동안 단계 710에서 수신된다. 이는 난수가 반복 누산 동안 한 번만 추가되게 한다. 난수는 단계 700에서 선형 귀환 이동 레지스터 또는 임의의 다른 방법에 의해 생성될 수 있다.
- [0057] 단계 715에서, 입력 수들 및 난수에 관해 반 가산이 수행된다. 숫자들은 어떠한 자리 올림도 전달하지 않고 합산되어 부분 합 출력을 생성하고, 생성된 자리 올림들은 별도의 자리 올림 출력으로서 집계된다. 일 실시 예에서, 단계들 705-715는 하나 이상의 자리 올림-보존 가산기에 의해 수행된다.
- [0058] 단계 720에서 부분 합 및 자리 올림 출력이 저장되고 귀환으로서 단계 705의 입력 수들로서 제공될 수 있다. 그러한 귀환은 하나 이상의 입력 수 스트림의 누산 부분 합산을 생성한다.
- [0059] 단계 725에서, 자리 올림 출력 및 부분 합이 수신되고, 단계 730에서, 부분 합 출력 및 자리 올림 출력의 전 가산(자리 올림 전달을 포함함)이 수행되어 입력 수(들)와 난수의 전 가산을 출력으로서 생성한다. 일 실시 예에서, 단계들 725-730은 가산기 로직에 의해 수행된다.
- [0060] 단계 735에서, 전 합산 출력의 최상위 비트들의 수가 둘 이상의 수의 합산의 추계학적으로 반올림된 결과로서 선택된다. 선택된 유효 비트들의 수는 난수에 사용되는 비트들의 수에 따른다.
- [0061] 도 8은 일 실시 예에서, 두 개 이상의 수의 부호 있는 합을 추계학적으로 반올림하기 위한 프로세스를 도시한 흐름도이다. 단계 805에서, 합산할 두 개 이상의 입력 수가 수신된다. 단계 810에서, 두 개 이상의 수의 전 가산(자리 올림 전달을 포함함)이 수행되어 중간 부호 있는 합이 생성된다. 단계 815에서 최상위 비트에 기초하여, 중간 합의 부호가 결정된다.
- [0062] 중간 합이 최상위 비트에 의해 셋팅된 것으로 표기되어 음수이면, 단계 820에서 난수가 음의 난수로 변환된다. 난수는 그것의 모든 비트를 부정하거나 난수의 최상위 비트만 부정함으로써 변환될 수 있다. 중간 합이 최상위 비트에 의해 클리어된 것으로 표기되어 양수이면, 난수는 변환 없이 사용된다.
- [0063] 단계 825에서 난수가 수신되고, 단계 830에서 단계 810에서 생성된 중간 합과 난수의 가산이 수행된다. 단계 830에서는 전 합산이 생성된다.
- [0064] 단계 835에서, 전 합산 출력의 최상위 비트들의 수가 둘 이상의 수의 합산의 추계학적으로 반올림된 결과로서 선택된다. 유효 비트들의 수는 난수 입력에 사용되는 비트들의 수에 따른다.

도면

도면1



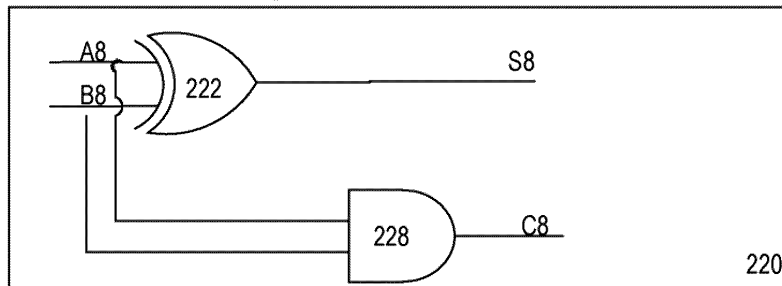
도면2



•

•

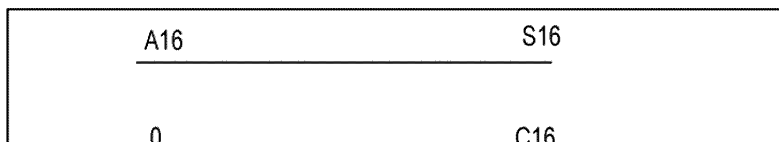
•



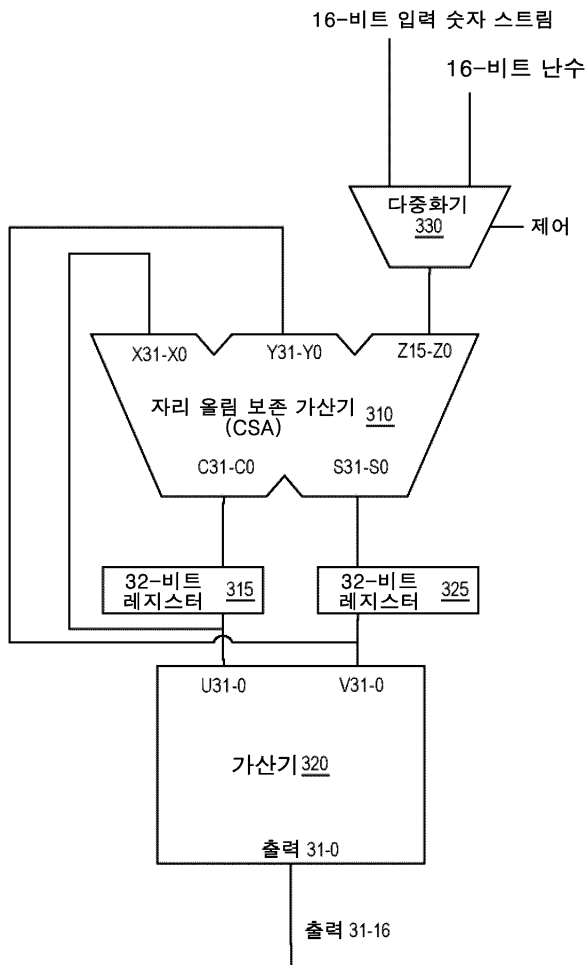
•

•

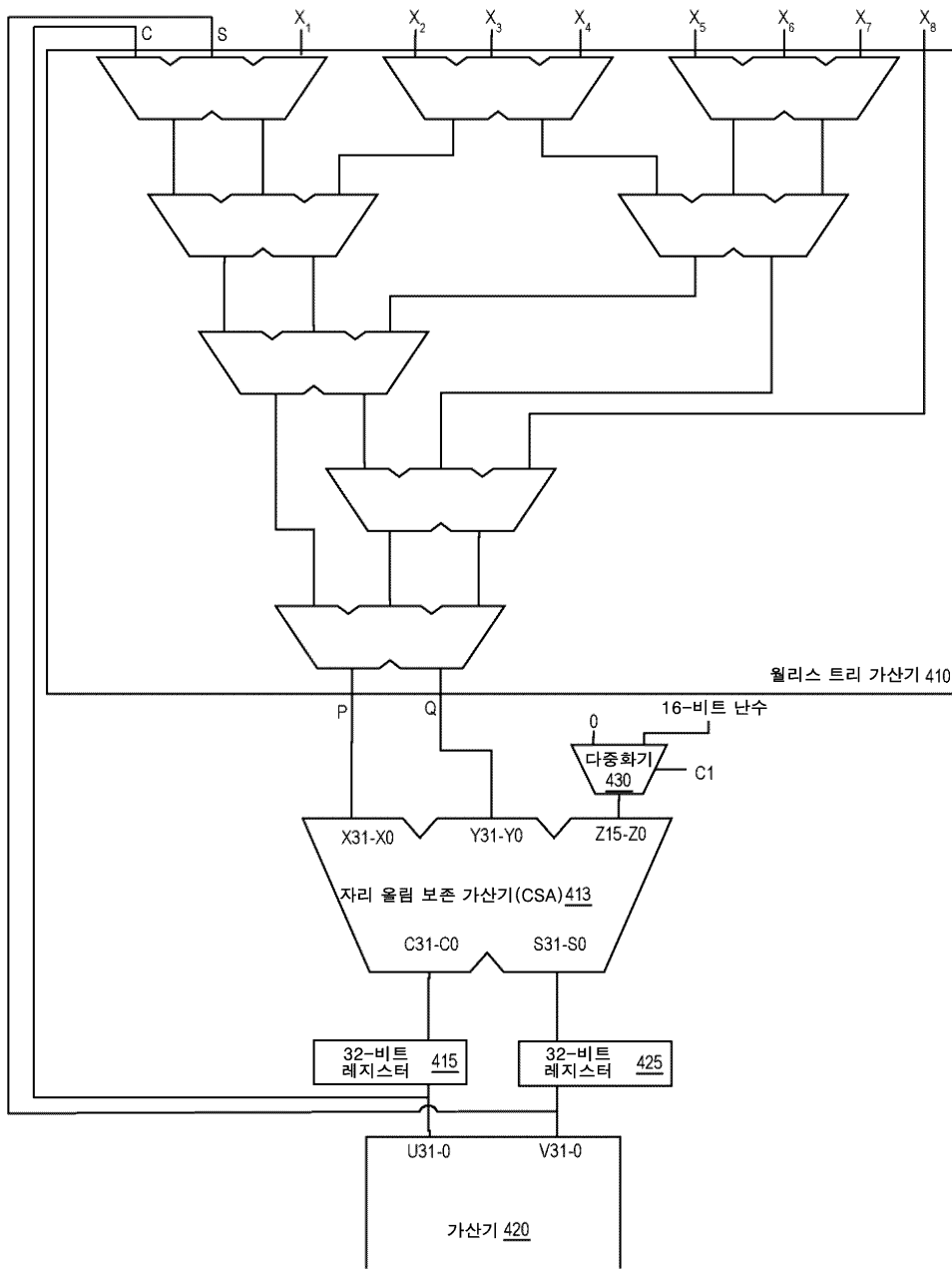
•



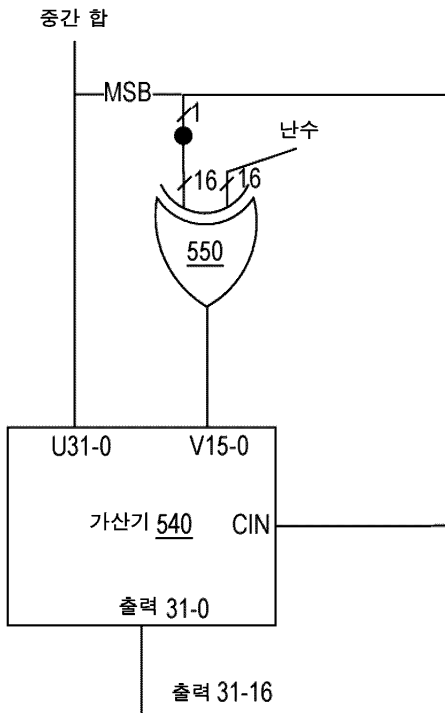
도면3



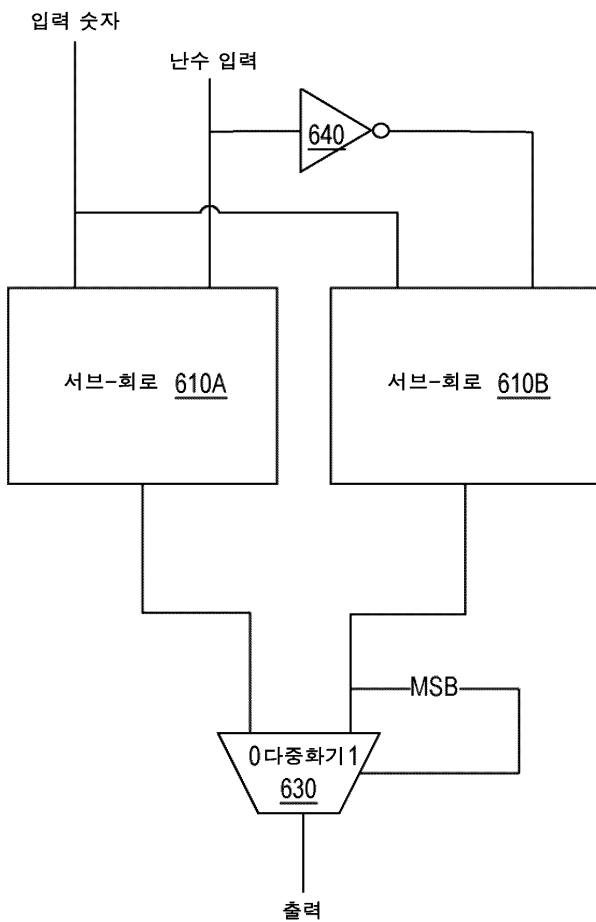
도면4



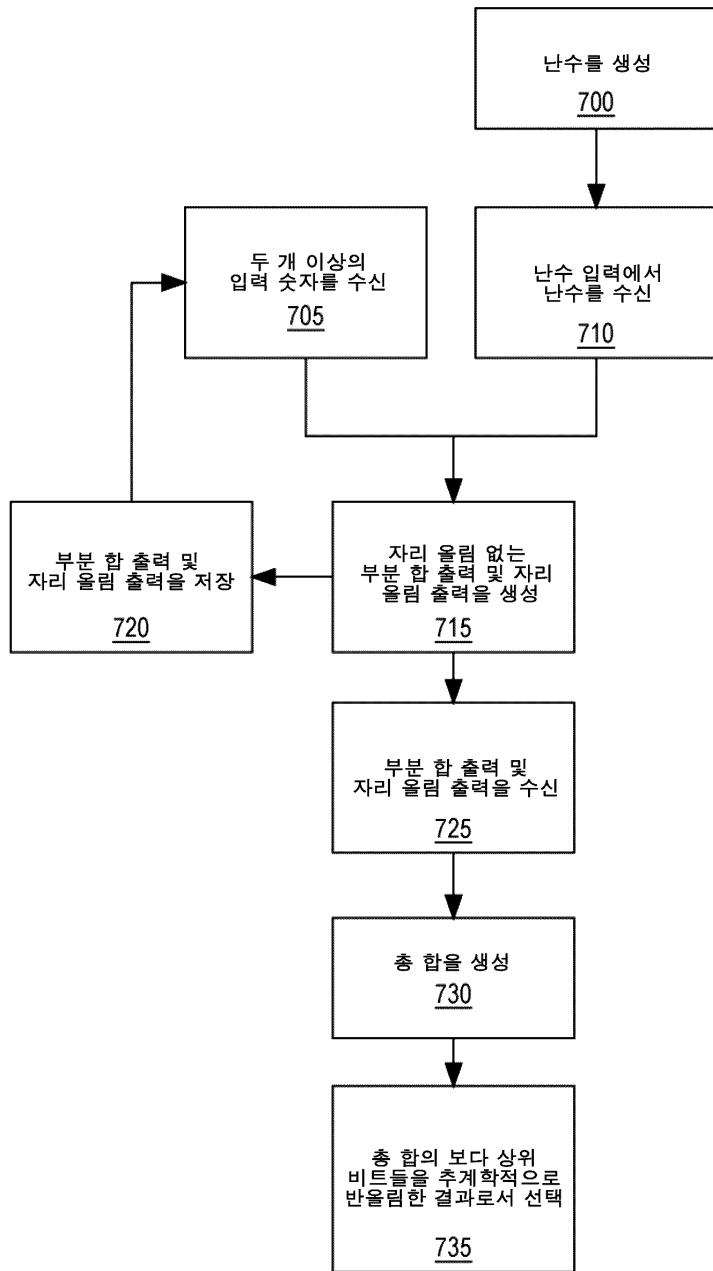
도면5



도면6



도면7



도면8

