

디지털 신호 처리 알고리즘의 특징은 거의 모든 경우에 연속적인 누산기능을 사용하고 있다는 것이다. 예로써, ① FIR(Finite Impulse Response) 필터는

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k)$$

라는 수식으로,

② 이산변환(Discrete Transform)은

$$X(n) = \sum_{k=0}^{N-1} x(k)W^{kn},$$

, 라는 수식으로,

$$W = \exp(-\frac{j2\pi}{N})$$

여기서,

③ 컨벌루션(Convolution)은

$$y(n) = h(n) * x(n) = \sum_{k=0}^{M-1} h(k)x(n-k)$$

여기서, $n = 0, 1, \dots, M-1$, $M = N_1+N_2-1$

로 표현할 수 있으며, 여기서 모두 $\sum_{k=0}^{N-1}$ 이라는 부분이 포함되어 있음을 알 수 있다.

누산 연산 동작 도중에 입력 데이터들의 연산결과가 때때로 저장할 수 있는 한계를 초과하는 경우가 발생되는 데, 이를 오버플로우가 발생되었다고 표현하며, 고정 소숫점 연산을 사용하는 경우 부호가 반전되는 경우가 발생된다.

예로써, 4bit 고정 소숫점(또는 정수) 연산에 있어서,

① 4bit로 표현할 수 있는 숫자 범위는 -8 ~ +7까지 이므로

(2의 보수를 사용하는 경우)

② 4+4=8과 같은 연산 결과를 나타낼 수 없으며,

③ 이 경우, 실질적으로는 0100 → +4

$$\frac{+0100}{1000} \begin{matrix} \rightarrow +4 \\ \rightarrow -8 \end{matrix}$$

즉, 4+4 = -8로 나타난다.

이러한 것은 연산에 있어서 아주 치명적인 오류로서 디지털 신호 처리시스템은 이러한 경우를 특별히 검출하여 가능한 범위내에서 오류값이 최소가 되도록 다음과 같은 방법을 사용하는데, 다음의 두가지 방법은 상호 배타적인 것이 아니므로 모두 적용하는 경우가 많다.

① 오버플로우를 감지하기 위한 별도의 플래그 비트 레지스터를 사용한다.

이 플래그 레지스터는 통상적으로 컨벌루션 도중에만 셋트되고, 리셋된다.

② 오버플로우가 발생된 경우, 연산결과를 덧셈결과의 반대부호의 최대수치가 되도록 한다. 예로써, 4+4=7이 되도록 하는데, 이것을 디지털 신호 처리기에 있어서는 포화(Saturation)라고 칭한다.

제1도는 일반적인 누산기의 블록도로서 이의 작용을 설명하면 다음과 같다.

가산기(11)는 새로운 입력 데이터(DATA IN)가 공급될때마다 그 입력 데이터(DATA IN)와 이전 스텝에서의 연산결과치 즉, 포화로직 처리부(13)로 부터 귀환 입력되는 데이터(DATA OUT)를 가산하여 출력하고, 이때, 오버플로우 검출부(12)는 상기 가산기(11)에서 오버플로우가 발생되었는지의 여부를 체크하여 그 결과를 포화로직 처리부(13)에 통보해준다.

이에 따라 상기 포화로직 처리부(13)는 상기 오버플로우 검출부(12)로부터 오버플로우 검출신호가 공급되지 않을때에는 상기 가산기(11)에서 출력되는 가산결과치(AR)를 그대로 최종 출력데이터(DATA OUT)로 출력하게 되나, 만약, 오버플로우 검출신호가 입력되면 그 가산결과치(AR)를 선택하여 출력하지 않고 허용된 비트내에서 반대부호의 최대수치를 연산하여 이를 최종 출력데이터(DATA OUT)로 출력하게 된다.

상기 포화로직 처리부(13)의 포화로직 처리과정을 제2도를 참조하여 좀더 상세히 설명하면 하기와 같다.

상기 가산기(11)에서 가산결과 오버플로우가 발생되지 않을때에는 오버플로우 검출부(12)로 부터 오버플로우 플래그입력(OF)이 0으로 공급되고, 이에 의해 멀티플렉서(22)는 상기 가산기(11)에서 출력되는 가산결과치(AR)를 그대로 최종 출력데이터(DATA OUT)로 출력하게 된다.

그러나, 가산기(11)에서 가산결과 오버플로우가 발생하는 경우에는 오버플로우 검출부(12)로부터 오버플로우 플래그 입력(OF)이 1로 공급되고, 이에 의해 멀티플렉서(22)는 반대부호 최대수치 연산기(21)에 의해 연산된 반대부호의 최대수치를 최종 출력데이터(DATA OUT)로 출력하게 된다.

예로써, 상기 가산결과치(AR)의 총 비트수가 1bit(MSB)의 부호비트와 3bit의 사인비트로 이루어지는 경우, 그 가산결과치(AR)가 111로 출력될때 상기 반대부호 최대수치 연산기(21)는 111 + 1을 연산하여 1000을 출력하게 된다.

그러나, 이와 같은 종래의 누산기에 있어서는 새로운 입력이 공급되어 누산동작이 이루어질때마다 반드시 오버플로우가 발생되는지를 검사하여 오버플로우가 발생된 것으로 판단될 때 허용된 비트내에서 반대부호의 최대수치를 연산하여 최종 출력데이터 출력하게 되므로 누산처리시간이 많이 소요되는 결함이 있었다.

따라서, 본 발명의 목적은 누산 도중에 오버플로우를 검출하지 않고, 부호비트를 확장한 상태로 원하는 횟수만큼 연산을 계속 수행한 후 그 연산된 최종값을 사용하고자 하는 단계에서 단 1회에 걸쳐 오버플로우를 검사하여 최종 출력을 발생하는 누산기를 제공함에 있다.

제3도는 상기의 목적을 달성하기 위한 본 발명 누산기의 일 실시 블록도로서 이에 도시한 바와 같이, 새로운 입력 데이터(DATA IN)의 비트수를 소정 비트수만큼 확장하여 출력하는 부호 확장기(31)와 ; 상기 부호 확장기(31)를 통해 입력데이터가 공급될때마다 그 입력데이터와 자신이 바로 이전 스텝에서 출력한 데이터를 가산하는 동작을 반복적으로 수행하는 가산기(32)와 ; 상기 가산기(32)에서 출력되는 가산결과치(AR)의 모든 비트수가 0이거나 1인지를 검출하여 그에 따른 검출신호를 출력하는 오버플로우 검출부(33)와 ; 최종 누산결과치의 출력요구가 발생될때, 상기 오버플로우 검출부(33)의 출력신호에 따라 상기 가산기(32)에서 출력되는 가산결과치(AR)를 바이패스시켜 최종 출력데이터(DATA OUT)로 출력하거나 현재 최상위비트(MSB)와 동일한 부호를 갖는 최대 수치를 최종 출력데이터(DATA OUT)로 출력하는 포화로직 처리부(34)로 구성된 것으로, 이와 같이 구성된 본 발명의 작용 및 효과를 첨부한 제4도를 참조하여 상세히 설명하면 다음과 같다.

새로운 입력 데이터(DATA IN)를 종래에서와 같이 가산기(32)의 일측 입력으로 직접 공급하지 않고 부호 확장기(31)를 통해 부호비트가 소정 비트수만큼 확장하여 공급되도록 하였는데, 이와 같이 하는 이유는 가능한 한 오버플로우의 발생을 억제할 수 있도록 하기 위함이다.

또한, 상기 가산기(31)는 상기 부호 확장기(31)를 통해 입력되는 데이터와 궤환입력되는 데이터를 가산함에 있어서, 종래와 같이 최종 출력데이터(DATA OUT)를 입력으로 하지 않고, 자신의 가산결과치(AR)를 그대로 선택하여 가산하게 된다.

즉, 상기 가산기(31)의 비트수는 상기 확장기(31)를 통해 확장된 부호비트수 만큼 확장되어 있으며, 오버플로우의 발생에 관계없이 이전 스텝에서 발생한 가산결과치(AR)를 선택하여 가산하게 된다.

이때, 제로비트 검출기(33A)는 상기 가산결과치(AR)를 나타내는 비트 중에서 확장부호 비트의 값을 계속적으로 체크하여 모두 0일때 소정의 검출신호(예 : 1)를 출력하고, 그 제로비트 검출기(33A)의 0 비트 검출동작과 병행하여 1비트 검출기(33B)는 확장부호 비트의 값을 계속적으로 체크하여 모두 1일때 소정의 검출신호(예 : 1)를 출력한다.

이에 따라 오아게이트(33C)에서는 상기 확장부호비트의 값이 모두 0이거나 1일때 1이 출력되지만 평상시에는 그 출력값이 포화로직 처리부(34)에 아무런 영향을 미치지 않는다.

하지만, 외부로부터 최종 누산결과치의 출력요구가 발생될때, 상기 포화로직 처리부(34)는 상기 오아게이트(33C)의 출력값을 확인하여 0이 입력되면 상기 가산기(32)에서 출력되는 가산결과치(AR)를 선택하여 그대로 최종 출력데이터(DATA OUT)로 출력하게 된다. 즉, 가산결과치(AR)를 바이패스시킨다.

그러나, 상기 오아게이트(33C)의 출력값이 1이면 즉, 가산결과치(AR)를 나타내는 비트 중에서 확장부호 비트의 값이 모두 0이거나 모두 1로 검출되는 경우, 상기 포화로직 처리부(34)가 구동되는데, 이때, 종래에서와 같이 반대부호의 최대수치를 최종 출력데이터(DATA OUT)로 출력하는 것이 아니라 게 된다.

여기서, 상기 포화로직 처리부(34)의 작용을 제4도를 참조하여 좀더 상세히 설명하면 하기와 같다.

오아게이트(33C)에서 0이 출력될때, 오버플로우 플래그입력(OF)이 0으로 공급되고, 이에 의해 멀티플렉서(42)는 상기 가산기(32)에서 출력되는 가산결과치(AR)를 그대로 최종 출력데이터(DATA OUT)로 출력하게 된다.

그러나, 상기 오아게이트(33C)에서 1이 출력될때, 오버플로우 플래그입력(OF)이 1로 공급되고, 이에 의해 상기 멀티플렉서(42)는 현재부호 최대치 연산기(41)에 의해 연산된 현재 부호의 최대수치를 선택하여 이를 최종 출력데이터(DATA OUT)로 출력하게 된다.

상기 현재부호 최대치 연산기(41)는 상기 가산결과치(AR)의 비트 중에서 버퍼(BUF11)를 통해 입력되는 최상위비트(MSB)와 인버터(111)를 통해 입력되는 N-1비트를 이용하여 현재 부호의 최대수치를 연산한다.

본 발명은 수회 이상의 연속적인 누산동작이 이루어지는 경우에 효과를 발휘하게 되는데, 그 이유를 설명하면 하기와 같다.

종래의 누산기는 매 가산(덧셈) 동작이 이루어질때마다 오버플로우를 검사하게 되는데, 이와 같은 동작은 통상적으로 불필요한 것이다. 즉, 연속적인 누산동작을 수행하는 도중 몇번째 가산에서 오버플로우가

발생되었는지는 별다른 의미가 없으며, 하나의 $\sum_{k=0}^{N-1}$ 로 표시되는 전체 누산과정에서 오버플로우가 발생되었는지의 여부를 확인하는 정도의 정보만으로도 충분하게 된다.

다시 말해서, 종래의 누산기에 있어서는 새로운 입력데이터가 공급될때마다 오버플로우 검출부와 포화로직 처리부가 동작하게 되어 그만큼 처리속도가 지연되는데, 본 발명에서는 이를 감안하여 최종 결과치를 출력할때에 한하여 오버플로우 검출부(33)와 포화로직 검출부(34)가 동작하도록 하고, 그 중간과정에서는 동작하지 않도록 하였다. 물론, 부호비트수 확장에 의하여 가산기의 비트수가 확장되어 있으므로 약간의 속도저하는 유발하게 되지만 그 정도가 포화로직 처리부(34)에 비하여 상대적으로 매우 작은 것이었 별다른 영향을 미치지않게 된다.

또한, 많은 경우에 있어서, 새로 입력되는 데이터의 가산결과 그 결과치의 부호가 +나 -로 임의로 변경될 수 있다. 이에 따라 최종 결과값은 오버플로우가 아니지만 중간의 연산단계에서는 종래와 같은 누산기를 사용하는 경우 오버플로우가 발생할 수 있다.

예로써, 4bit 누산기에 있어서, 세번에 걸쳐 입력되는 데이터값이 각각 +4, +4, -2인 경우 누산결과 +6이 아닌 +5가 될 것이다.

왜냐하면, 4+4 ⇒ -8 : 오버플로우

⇒ 7 : 포화

∴ 7 + (-2) ⇒ 5

그러나, 본 발명에서는 이러한 경우에 상기의 설명에서와 같이 중간의 누산과정에서 오버플로우를 처리하지 않기 때문에 +6을 최종적인 결과값으로 처리할 수 있게 된다.

한편, 제5도는 본 발명에 의한 부호비트 확장을 보인 것으로 이에 도시한 바와 같이, 32bit의 입력데이터에 부호 확장기(51)를 통해 4bit를 확장하여 총 36bit 출력을 발생하도록 하고, 그 중에서 5bit를 확장부호로 사용하도록 하였으며, 이의 전반적인 동작과정은 제3도와 동일하므로 별도의 설명은 생략한다. 통상적으로, 부호 확장범위는 2~8bit가 적당하지만 경우에 따라 그 이외의 범위도 사용할 수 있는 것이다.

이상에서 상세히 설명한 바와 같이, 본 발명은 종래에서와 같이 새로운 입력데이터가 공급될때마다 오버플로우 검출부와 포화로직 처리부가 동작하지 않고 최종 결과치를 출력할때에 한하여 동작하도록 하고 그 중간과정에서는 동작하지 않도록 함으로써 누산기의 처리속도를 향상시킬 수 있는 효과가 있고, 오히려 중간의 누산과정에서 오버플로우를 처리하지 않기 때문에 보다 정확한 결과치를 산출할 수 있는 효과가 있다.

(57) 청구의 범위

청구항 1

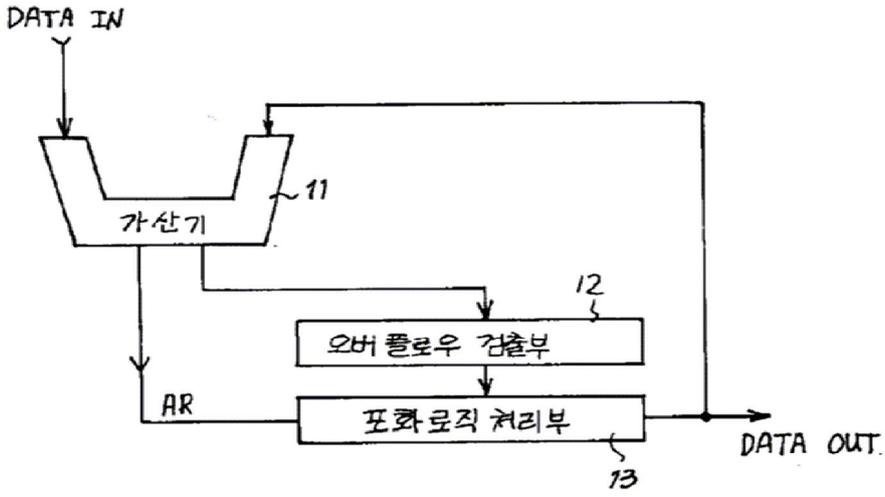
새로운 입력 데이터(DATA IN)의 비트수를 소정 비트수만큼 확장하여 출력하는 부호 확장기(31)와 ; 상기 부호 확장기(31)를 통해 입력데이터가 공급될때마다 그 입력데이터와 자신이 바로 이전 스텝에서 출력한 데이터를 가산하는 동작을 반복적으로 수행하는 가산기(32)와 ; 상기 가산기(32)에서 출력되는 가산결과치(AR)의 모든 비트수가 0이거나 1인지를 검출하여 그에 따른 검출신호를 출력하는 오버플로우 검출부(33)와 ; 최종 누산결과치의 출력요구가 발생될때, 상기 오버플로우 검출부(33)의 출력신호에 따라 상기 가산기(32)에서 출력되는 가산결과치(AR)를 최종 출력데이터(DATA OUT)로 출력하거나 현재 최상위비트(MSB)와 동일한 부호를 갖는 최대 수치를 최종 출력데이터(DATA OUT)로 출력하는 포화로직 처리부(34)로 구성된 것을 특징으로 하는 누산기.

청구항 2

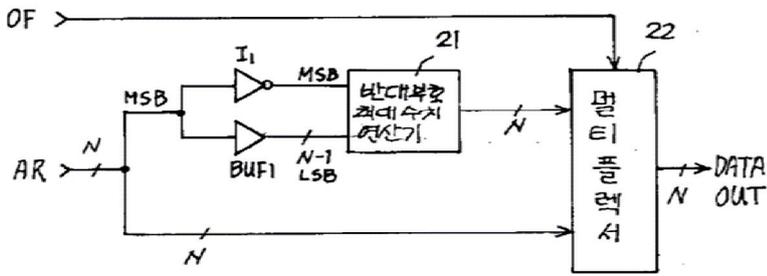
제1항에 있어서, 오버플로우 검출부(33)는 상기 가산결과치(AR)를 나타내는 비트 중에서 확장부호 비트의 값을 계속적으로 체크하여 모두 0일때 하이신호를 출력하는 제로비트 검출기(33A)와 ; 상기 가산결과치(AR)를 나타내는 비트중에서 확장부호 비트의 값을 계속적으로 체크하여 모두 1일때 하이신호를 출력하는 1비트 검출기(33B)와 ; 상기 제로비트 검출기(33A)의 출력신호와 1비트 검출기(33B)의 출력신호를 오아연산하여 그 결과를 상기 포화로직 처리부(34)로 출력하는 오아게이트(33C)로 구성된 것을 특징으로 하는 누산기.

도면

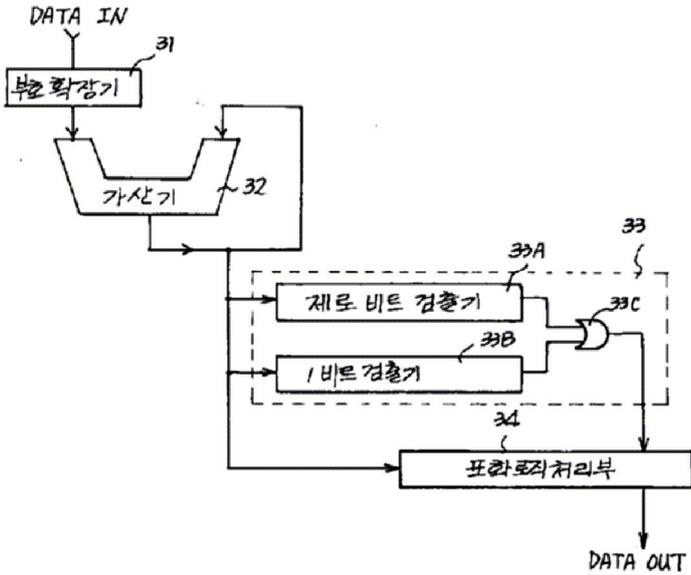
도면1



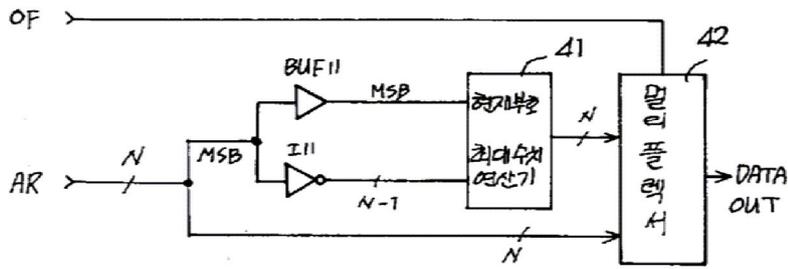
도면2



도면3



도면4



도면5

