(19) 대한민국특허청(KR)
(12) 특허공보(B1)

(51) Int. Cl.[6]
G06F 7/52

(45) 공고일자 1995년06월19일
(11) 공고번호 특1995-0006580

(21) 출원번호	특1988-0003423	(65) 공개번호	특1989-0015121
(22) 출원일자	1988년03월29일	(43) 공개일자	1989년10월28일

(30) 우선권주장 62-79158 1987년03월31일 일본(JP)

(71) 출원인 가부시키가이샤 도시바 아오이 죠이치
일본국 가나가와현 가와사키시 사이와이구 호리가와정 72번지

(72) 발명자 미요시 아키오
일본국 가나가와현 가와사키시 사이와이구 고무가이 도시바정 1번지 가부시키가이샤 도시바 다마가와공장내

(74) 대리인 이범일, 김윤배

심사관 : 오홍수 (책자공보 제4014호)

(54) 나눗셈연산장치

요약

내용 없음.

대표도

도1

명세서

[발명의 명칭]

나눗셈연산장치

[도면의 간단한 설명]

제1도는 본 발명의 1실시예에 따른 나눗셈연산장치의 블록도.

제2도는 제1도에 도시된 장치의 동작을 설명하기 위한 플로우차트.

제3도는 종래의 나눗셈연산장치의 블록도.

제4도는 제3도에 도시된 장치의 동작을 설명하기 위한 플로우차트이다.

* 도면의 주요부분에 대한 부호의 설명

1 : 제1레지스터 2 : 제2레지스터

3 : 제3레지스터 4 : ALU

5 : 나눗셈결과생성회로 6 : ALU 기능 선택회로

7 : 오버플로우 지시회로 S1~S2 : 플로우챠트의 각 스텝

[발명의 상세한 설명]

[산업상의 이용분야]

본 발명은 나눗셈연산장치(除算演算裝置)에 관한 것으로, 특히 감산과 시프트동작을 반복하여 몫(商)을 구하는 나눗셈연산장치에 관한 것이다.

[종래의 기술 및 그 문제점]

나눗셈연산장치에서는 피제수(被除數)를 저장하기 위한 레지스터와, 제수를 저장하기 위한 레지스터 및, 몫을 저장하기 위한 레지스터가 사용되고 있는데, 일반적으로 피제수를 저장하기 위한 레지스터의 일부를 몫을 저장하기 위한 레지스터에 겸용하고 있다. 그리고, 나눗셈은 피제수의 상위 자릿수로부터 제수를 감산하고 1회의 감산종료후에 피제수를 좌측으로 시프트시켜서 다음의 감산을 수행하는 조작을 반복함으로써 실행되는 바, 이러한 나눗셈에서 몫은 각 감산결과의 부호를 1 및 0으로 표

현하여 나란히 배열함으로써 얻어지게 된다.

그런데, 피재수가 제수에 비해 매우 크면, 나눗셈 결과 얻어지는 몫도 커지게 되어 그 몫을 저장하기 위한 자릿수가 부족해지는 사태가 생긴다. 즉, 나눗셈이 오버플로우(overflow)되어 버려서 정확한 연산결과를 얻을 수 없게 된다. 이 때문에, 종래의 나눗셈연산장치에서는 실제의 나눗셈연산을 하기 전에 피제수와 제수를 비교하는 비교연산을 하여 오버플로우인가 아닌가를 판정하고 있다.

상기한 바와 같이 종래의 나눗셈연산장치에서는 모든 나눗셈연산 이전에 오버플로우이나 아닌가를 조사하는 비교연산을 하고 있기 때문에 여분의 시간이 더 걸리게 되어 연산속도가 느려진다는 문제가 있었다.

[발명의 목적]

본 발명은 상기한 점을 감안해서 발명된 것으로, 별도의 비교연산을 할 필요가 없어서 연산속도를 빠르게 할 수 있는 나눗셈연산장치를 제공하고자 함에 그 목적이 있다.

[발명의 구성]

상기 목적을 달성하기 위한 본 발명은, 피제수의 상위자리를 저장하기 위한 제1레지스터와, 상기 피제수의 하위자리를 저장하기 위한 제2레지스터, 제수를 저장하기 위한 제3레지스터, 상기 제1레지스터의 내용과 상기 제3레지스터의 내용간에는 감산 또는 가산을 실행하는 ALU , 이 ALU의 연산결과가 부(-)로 된 것을 나타내는 나눗셈결과생성회로 및 , 이 나눗셈결과생성회로가 부(-)를 나타내고 있을 때에만 상기 ALU에서 가산을 수행하게 하는 ALU기능선택회로를 구비하고, 상기 ALU가 1회의 연산을 종료한 후, 그 연산결과를 1비트만큼 좌측으로 시프트시켜 상기 제1레지스트에 저장하고, 상기 제2레지스터의 내용을 1비트만큼 좌측으로 시프트시킴과 더불어 그 최상위 비트를 상기 제1레지스터의 최하위 비트에 저장하며, 상기 나눗셈생성회로가 부(-)를 나타내고 있는때에는 "0"을, 그 이외에는 "1"을 상기 제2레지스터의 최하위 비트에 저장하여 다음 연산을 실행하고, 이하 상기 제1레지스터에서 몫이 얻어지기까지 필요한 회수만큼 연산동작을 계속하도록 구성된 나눗셈연산장치에
있어서, 몫을 얻기에 필요한 횟수의 연산종료후에 상기 제2레지스터의 최상위 비트로부터 시프트되어 오는 비트를 유지하여 이 비트가 "1"인 경우에 나눗셈이 오버플로우상태인 것을 나타내게 되는 오버플로우 지시회로를 설치한 것을 특징으로 한다.

[작용]

상기와 같이 구성된 본 발명에 따른 나눗셈연산장치에서는, 필요한 횟수째의 연산종료후에 제2레지스터의 최상위 비트로부터 시프트되어 오는 비트에 의해 오버플로우인가 아닌가가 판단되는 바, 즉 이 비트가 "1"인 경우에 제산이 오버플로우라는 것이 표시된다. 이 비트는 최초의 감산을 실행한 때의 연산결과에 의해 나눗셈결과생성회로에서 생성된 것으로, 이 비트가 "1"인 것은 최초의 감산결과가 정(+)이라는 것을 나타낸다. 즉, 최초의 시점에서 제1레지스터의 내용이 제3레지스터의 내용보다 크든가 같다는 것을 나타낸다. 이와 같이 나눗셈연산의 최후에 오버플로우인가 아닌가를 판정할 수 있기 때문에, 종래의 나눗셈연산장치와 같이 나눗셈연산전에 비교연산을 행할 필요가 없게 되어, 여분의 연산을 하지 않고 연산속도를 향상시킬 수 있게 된다.

[실시예]

이하, 예시도면을 참조하여 본 발명에 따른 실시예를 상세히 설명한다.

본 발명에 따른 나눗셈연산장치는 종래의 나눗셈연산장치에 새로운 오버플로우 지시회로를 설치한 점에 특징이 있다. 이하의 설명에서는 편의상 우선 종래의 나눗셈연산장치의 구성과 그 동작을 참고로 설명한다.

제3도는 종래의 나눗셈연산장치의 일례를 나타낸 블록도로서, 이 장치는 3개의 레지스터(1,2,3)를 갖추고 있는데, 각 레지스터는 예컨대 8비트의 데이터를 보존할 수 있는 것이다. 여기에서 피제수의 상위자리는 제1레지스터(1)에, 하위자리는 제2레지스터(2)에, 제수는 제3레지스터(3)에 저장된다. 그리고, 나눗셈연산의 결과인 몫은 최종적으로 제2레지스터(2)에서 얻어진다.

따라서, 본 예에서는 16비트까지의 피제수를 8비트까지의 제수로 나누는 연산을 할 수 있는데, 그 결과 얻어지는 몫이 8비트를 넘는 경우에 오버플로우로 된다.

이하, 설명을 간단하게 하기 위하여 제1레지스터(1)와 제2레지스터(2) 및 제3레지스터(3)의 내용을 각각 데이터(A), 데이터(C), 데이터(B)로 부르기로 한다.

ALU(4)는 데이터(A)와 데이터(B)간의 각종 연산을 하는 회로인데, 여기에서는 양 데이터의 감산과 가산외에 비교연산이 실행된다. 그러한 감산 및 가산의 결과를 다시 제1레지스터(1)에 돌아오게 되는 바, 이때에는 후술하겠지만 좌로 1비트분 시프트시키고 나서 돌아오게 된다. 또한, 나눗셈결과생성회로(5)는 ALU(4)에서 실행된 연산에 따라 "0" 또는 "1"을 출력한다. 즉, 연산결과가 부(-)인때에는 "0"이, 그 이외인 때에는 "1"이 출력된다. 이 출력은 제2레지스터(2)의 최하위 비트에 저장된다. ALU 기능선택회로(6)는 나눗셈연산중에 ALU(4)가 실행하는 연산을 감산으로 할 것인가 가산으로 할 것인가를 지정하는 회로인 바, 나눗셈결과생성회로(5)로부터 "0"이 부여되며 ALU(4)의 다음회의 연산을 가산으로 지정하고, "1"이 부여되면 ALU(4)의 다음회의 연산을 감산으로 지정하는 기능을 수행한다.

이하, 상기와 같은 장치에 의한 나눗셈의 수순을 제4도에 도시된 플로우차트에 의거 설명한다.

먼저, 스텝(S1)에서 제수를 데이터(B)로서 레지스터(3)에 설정한다. 이어 스텝(S2)에서 피제수의 상위자리를 데이터(A)로서 레지스터(1)에, 하위자리를 데이터(C)로서 레지스터(2)에 설정한다.
그리고, 스텝(S3)에서 ALU(4)를 비교연산으로 설정한다. 이에 따라 스텝(S4)에서 데이터(A)와 데이터(B)의 비교연산이 이루어진다. 여기에서 A≥B인 경우에는 스텝(S5)에서 오버플로우처리가 이루어져 나눗셈은 실행되지 않는다. 상기 A≥B라 하는 것은 피제수의 상위자리가 제수보다 크므로 나눗셈

을 실행해도 몫의 자릿수가 많게 되어, 제2레지스터(2)에 그 몫을 저장할 수 없다는 것을 뜻한다.

상기 스텝(S4)에서 A<B인 경우에만 이하의 나눗셈이 실행되게 된다. 즉, 우선 스텝(S6)에서 ALU(4)는 감산으로 설정된다. 따라서, 스텝(S7)에서는 A-B의 감산이 실행되고, 그 연산결과 얻어지는 데이터는 좌로 1비트만큼 시프트되어 제1레지스터(1)에 저장된다.

그리고 스텝(S8)에서는 제2레지스터(2)내의 데이터(C)를 좌측으로 1비트만큼 시프트시킨다. 이때 제2레지스터(2)로부터 나오는 데이터(C)의 최상위 비트는 제1레지스터(1)의 최하위 비트에 저장되게 된다.

또한, 상기 제2레지스터(2)의 최하위 비트에는 스텝(S9)에서 나눗셈결과생성회로(5)의 출력이 저장되게 된다. 상기 나눗셈결과생성회로(5)는 상기한 바와 같이 ALU(4)에서의 연산결과가 부(-)인 경우에 "0"을, 그이외의 경우에 "1"을 각각 발생시킨다. 결국, 이 나눗셈 결과생성회로(5)가 발생시킨 비트열이 몫(商)으로 된다.

또한, 스텝(S10)에서 ALU 기능선택회로(6)가 나눗셈 결과생성회로(5)의 출력에 기초하여 ALU(4)에서 다음번에 수행되는 연산을 감산으로 할 것인가 가산으로 할 것인가를 설정한다. 여기에서 나눗셈 결과생성회로(5)가 "0"을 출력하고 있는 때에는 가산을, "1"을 출력하고 있는 때에는 감산을 각각 설정하게 된다.

이상에서 설명한 스텝(S7~S10)의 수순이 소정 회숫, 즉 제2레지스터(2)의 비트수 + 1회만큼 반복된다. 또한, 이때 스텝(S7)의 연산은 스텝(S10)에서 설정된 연산을 실행하게 된다. 이렇게 해서 소정 회수만큼의 반복이 종료도고 나서 스텝(S11)을 거쳐 모든 나눗셈수순이 종료되고, 구해내야 할 몫은 제2레지스터(2)에서 얻어지게 된다.

상기한 종래 기술에 대응되게 제1도에는 본 발명에 따른 나눗셈연산장치의 1실시예가 블록도로 도시되어 있는데, 본 장치는 제3도에 도시된 종래 장치에다가 오버플로우 지시회로(7)를 더 설치한 것이다. 본 실시예에서 오버플로우 지시회로(7)는 제2레지스터(2)내의 데이터(C)의 캐리플래그(carry flag)를 나타내는 것이다.

따라서, 제2레지스터(2)내의 데이터(C)가 좌측으로 시프트된 때의 본래의 최상위 비트는 제1레지스터(1)의 최하위 비트에 저장됨과 더불어 오버플로우 지시회로(7)내에 캐리플래그로서 보존되게 된다.

상기한 장치에 의한 나눗셈의 수순을 제2도의 플로우차트에 도시하였는 바, 스텝(S1) 및 스텝(S2)은 종래의 수순과 모두 동일하다. 그런데, 종래의 스텝(S3,S4)은 실행됨이 없이 스텝(S6)으로부터의 나눗셈수순이 시작된다. 즉, 미리 오버플로우인가 아닌가를 확인하지 않고 나눗셈수순이 진행되게 된다. 한편, 스텝(S6)~스텝(S11)까지의 몫을 구하는 반복수순은 종래의 수순과 거의 같지만, 단 스텝(S8')에 나타낸 바와 같이 데이터(C)를 시프트시키는 수순에서 데이터(C)의 최상위 비트는 제1레지스터(1)의 최하위 비트에 저장됨과 더불어 오버플로우 지시회로(7)에 보존된다.

위와 같이 소정 횟수의 반복이 종료되면, 스텝(S12)에서 오버플로우 지시회로(7)의 내용에 대한 점검이 실행되는 바, 이 오버플로우 지시회로(7)내에 보존되어 있는 1비트의 플래그는 본래 제1회째의 ALU(4)에서의 연산결과가 부(-)인가 아닌가를 나타내는 플래그이다. 즉, 제1회째의 ALU(4)에 의한 연산종료시에 나눗셈 결과생성회로(5)로부터 출력된 1비트가 제2레지스터(2)에서의 소정 횟수의 시프트동작에 의해 최종적으로 오버플로우 지시회로(7)까지 시프트된 것이다. 따라서, 스텝(S12)에서 이 플래그가 "1"이라고 판단되면, 제1회째의 ALU(4)에 의한 연산결과가 부(-)가 아닌 바, 바꾸어 말하면 최초의 시점에서 A≥B인 것으로 된다.

여기에서 이 경우에는 제2레지스터(2)에 최종적으로 얻어진 데이터(C)는 몫으로 올바른 값이 아니므로, 스텝(S5)에서 오버플로우의 처리가 실행된다. 역으로, 오버플로우 지시회로(7)내의 플래그가 "0"이라면, 데이터(C)에 올바른 몫이 얻어진 것으로 된다.

상기한 바와 같이 종래의 나눗셈연산장치의 연산에 있어서 스텝(S3,S4)에서 이루어지는 오버플로우의 판단수순을 나눗셈수순의 앞에서 꼭 실행할 필요는 없게 되는 바, 즉 본 발명에 따른 나눗셈연산장치에서는 나눗셈수순 종료후에 스텝(S12)에서 오버플로우 지시회로(7)내의 플래그를 확인하기만 하면 된다. 이 때문에 여분의 연산을 생략할 수 있으므로 연산시간이 단축되게 된다.

한편, 본원 청구범위의 각 구성요소에 병기한 도면참조부호는 본원 발명의 이행를 용이하게 하기 위한 것으로, 본원 발명의 기술적 범위를 도면에 도시한 실시예로 한정할 의도로 병기한 것은 아니다.

[발명의 효과]

이상 설명한 본 발명의 나눗셈연산장치에 의하면, 최종적으로 몫이 얻어지는 레지스터로부터 최후로 시프트되어 온 1비트의 데이터에 의해 오버플로우의 유무를 판단하도록 하였기 때문에, 여분의 연산을 생략할 수 있고, 이로써 연산속도를 향상시킬 수 있게 된다.

(57) 청구의 범위

청구항 1

피제수의 상위자리를 저장하기 위한 제1레지스터(1)와, 상기 피제수의 하위자리를 저장하기 위한 제2레지스터(2), 제수를 저장하기 제3레지스터(3), 상기 제1레지스터(1)의 내용과 상기 제3레지스터(3)의 내용간에서 감산 또는 가산을 실행하는 ALU(4), 이 ALU(4)의 연산결과가 부(-)로 된 것을 나타내는 나눗셈결과생성회로(5) 및, 이 나눗셈결과생성회로(5)가 부(-)를 나타내고 있는 때에만 상기 ALU(4)에서 가산을 수행하게 하는 ALU 기능선택회로(6)를 구비하고, 상기 ALU(4)가 1회의 연산을 종료한 후, 그 연산결과를 1비트만큼 좌측으로 시프트시켜 상기 제1레지스터(1)에 저장하고, 상기 제2레지스터(2)의 내용을 1비트만큼 좌측으로 시프트시킴과 더불어 그 최상위 비트를 상기 제1레지스터

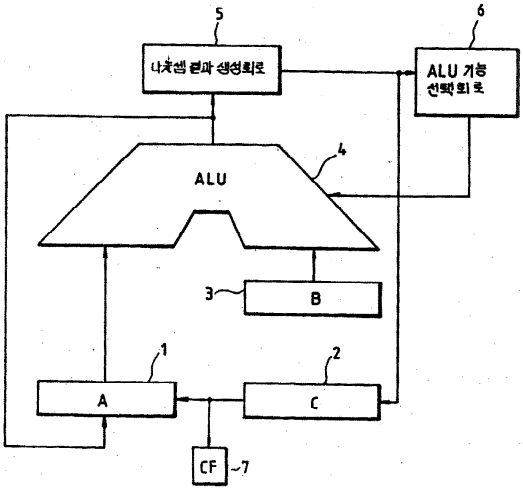
(1)의 최하위 비트에 저장하며, 상기 나눗셈결과생성회로(5)가 부(-)를 나타내고 있는 때에는 "0"을, 그 이외에는 "1"을 상기 제2레지스터(2)의 최하위 비트에 저장하여 다음 연산을 실행하고, 이하 상기 제2레지스터(2)에서 몫이 얻어지기까지 필요한 횟수만큼 연산동작을 계속하도록 구성된 나눗셈연산장치에 있어서, 몫을 얻기에 필요한 횟수의 연산종료후에 상기 제2레지스터(2)의 최상위 비트로부터 시프트되어 오는 비트를 유지하여 이 비트가 "1"인 경우에 나눗셈이 오버플로우상태인 것을 나타내게 되는 오버플로우 지시회로(7)를 설치한 것을 특징으로 하는 나눗셈연산장치.

청구항 2

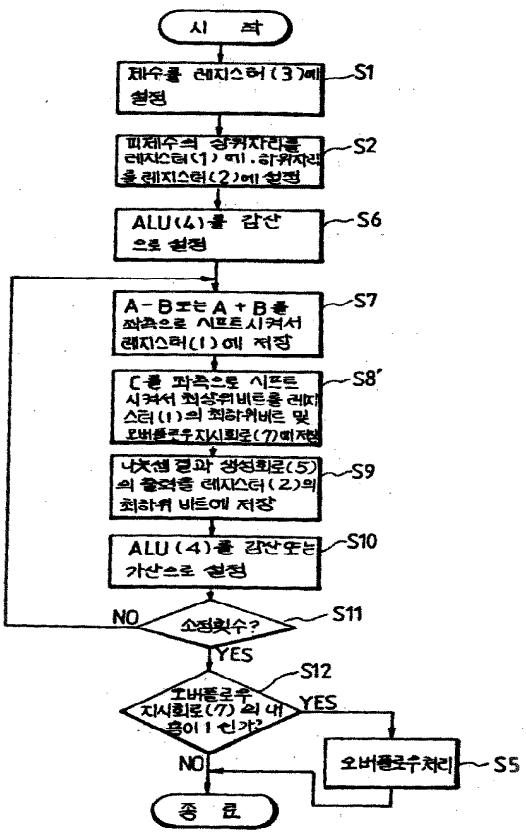
제1항에 있어서, 제1레지스터(1)와 제2레지스터(2) 및 제3레지스터(3)가 각각 동일한 비트용량을 갖고 있는 것을 특징으로 하는 나눗셈연산장치.

도면

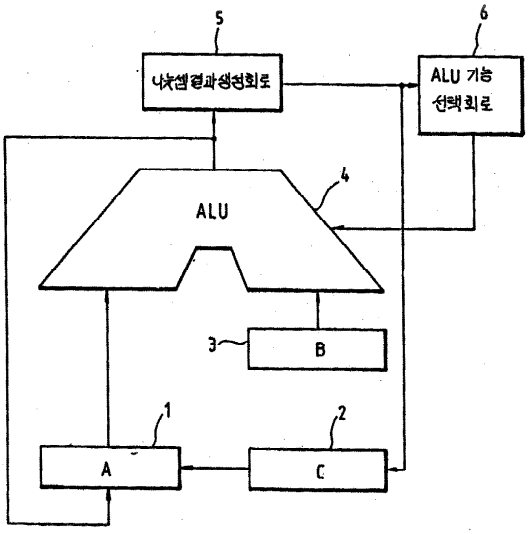
도면1



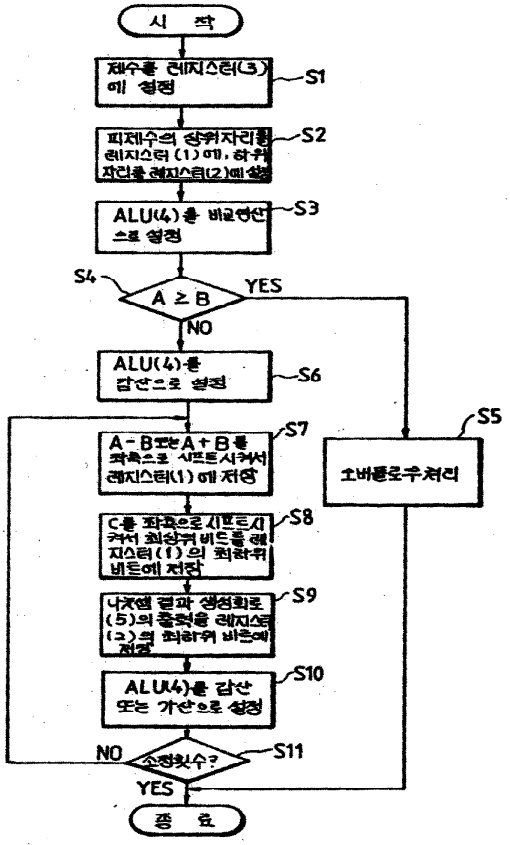
도면2



도면3


5
나눗셈결과생성회로
6
ALU 기능
선택회로
4
ALU
3
B
1
2
A
C

도면4


시 작
제수를 레지스터(3)에 설정
S1
피제수의 상위자리를 레지스터(1)에, 하위자리를 레지스터(2)에 설정
S2
ALU(4)를 비교연산으로 설정
S3
S4
A ≥ B
YES
NO
ALU(4)를 감산으로 설정
S6
S5
A-B또는A+B를 좌측으로 시프트시켜서 레지스터(1)에 저장
S7
소바룰로주처리
C를 좌측으로 시프트시켜서 최상위비트를 레지스터(1)의 최하위비트에 저장
S8
나눗셈 결과 생성회로(5)의 출력을 레지스터(2)의 최하위 비트에 저장
S9
ALU(4)를 감산 또는 가산으로 설정
S10
NO
소정횟수?
S11
YES
종 료