



(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(45) 공고일자 2008년04월16일  
(11) 등록번호 10-0821616  
(24) 등록일자 2008년04월04일

(51) Int. Cl.

G06F 7/06 (2006.01) G06F 7/02 (2006.01)

G06F 7/04 (2006.01)

(21) 출원번호 10-2006-0127879

(22) 출원일자 2006년12월14일

심사청구일자 2006년12월14일

(56) 선행기술조사문헌

JP01142824 A

(73) 특허권자

한국과학기술원

대전 유성구 구성동 373-1

(72) 발명자

유희준

대전광역시 유성구 구성동 한국과학기술원 5-1221

김주영

대전 유성구 구성동 한국과학기술원 전자전산학과  
유희준교수님연구실

(74) 대리인

허진석

전체 청구항 수 : 총 6 항

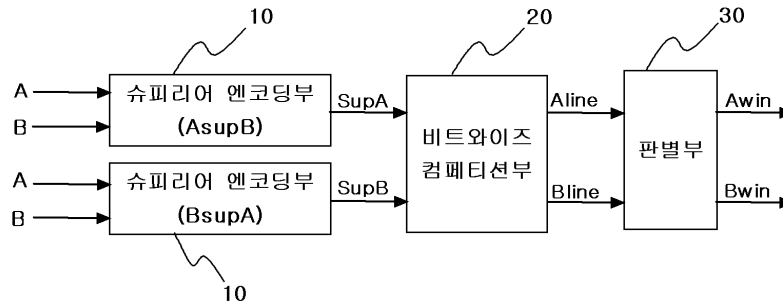
심사관 : 한선경

(54) 반도체 집적회로의 두 숫자 비교기

(57) 요약

본 발명은 반도체 집적회로의 두 숫자 비교기에 관한 것으로서, 반도체 집적회로에서 긴 비트의 두 숫자의 크기를 비교할 때 뺄셈기에 의한 연산 결과를 통해 판단하지 않고 두 숫자의 각 비트를 MSB로부터 비트단위로 각각 비교하여 먼저 1이 나오는 쪽의 숫자를 큰 수로 결정함으로써 트랜지스터 개수를 줄여 효율적인 크기 비교연산을 수행할 수 있는 이점이 있다.

대표도 - 도1



**특허청구의 범위**

**청구항 1**

어느 한 숫자의 각 비트에 대응되는 다른 한 숫자의 각 비트에 대해 우위인 비트를 변환하여 출력하는 슈퍼리어 엔코딩부와,

비교하기 위한 두 숫자에 대해 각각 슈퍼리어 엔코딩부에서 변환되어 출력된 두 우위값의 각 비트를 최상위비트로부터 순차적으로 각 비트를 비교하여 각 비트의 크기를 결정하여 출력하는 비트와이즈 컴페디션부와,

상기 비트와이즈 컴페디션부에서 순차적으로 각 비트의 크기가 결정되어 출력되는 값을 입력받아 크기가 결정된 최초의 상위 비트에 따라 두 숫자의 크기를 판별하는 판별부

를 포함하여 이루어진 것을 특징으로 하는 반도체 집적회로의 두 숫자 비교기.

**청구항 2**

제 1항에 있어서, 상기 슈퍼리어 엔코딩부는 어느 한 숫자를 반전하여 다른 한 숫자와 노어케이팅하는 것을 특징으로 하는 반도체 집적회로의 두 숫자 비교기.

**청구항 3**

제 1항에 있어서, 상기 슈퍼리어 엔코딩부에서 두 비트 중 우위인 비트는 '1'로 변환되고 동일한 비트는 '0'으로 변환되는 것을 특징으로 하는 반도체 집적회로의 두 숫자 비교기.

**청구항 4**

제 1항에 있어서, 상기 비트와이즈 컴페디션부에서 두 숫자의 최상위 비트는 부호비트가 아닌 것을 특징으로 하는 반도체 집적회로의 두 숫자 비교기.

**청구항 5**

제 1항에 있어서, 상기 비트와이즈 컴페디션부에서 최상위 비트가 부호비트인 숫자체계에서는 두 숫자의 최상위 비트를 맞바꾸어 판단하는 것을 특징으로 하는 반도체 집적회로의 두 숫자 비교기.

**청구항 6**

제 1항에 있어서, 상기 판별부는 크기가 결정된 최초의 상위 비트 값을 고정하여 출력하는 것을 특징으로 하는 반도체 집적회로의 두 숫자 비교기.

**명세서**

**발명의 상세한 설명**

**발명의 목적**

**발명이 속하는 기술 및 그 분야의 종래기술**

- <10> 본 발명은 반도체 집적회로의 두 숫자 비교기에 관한 것으로서, 보다 상세하게는 반도체 집적회로에서 긴 비트의 두 숫자의 크기를 비교할 때 뺄셈기에 의한 연산 결과를 통해 판단하지 않고 두 숫자의 각 비트를 MSB로부터 비트단위로 각각 비교하여 먼저 1이 나오는 쪽의 숫자를 큰 수로 결정함으로써 트랜지스터 개수를 줄여 효율적인 연산을 수행할 수 있도록 하는 반도체 집적회로의 두 숫자 비교기에 관한 것이다.
- <11> 일반적으로 이미지 처리나 그래픽 처리 등의 분야에서 두 숫자의 크기를 비교하는데 있어 뺄셈기를 이용하여 'A-B' 연산을 수행한 후, 양의 값이 나오면 A가 큰 값, 음의 값이 나오면 B가 큰 값, '0'이 나오면 같은 값으로 결정하였다.
- <12> 하지만 이런 방법은 단순히 두 숫자의 크기를 비교하는 목적에 비하여 두 수의 연산을 위한 트랜지스터의 수가 많이 필요하기 때문에 불필요한 정보도 많이 얻게 되어 비효율적인 측면이 있을 뿐만 아니라 여러 숫자의 크기 비교를 많이 수행하게 되는 위의 이미지 처리나 그래픽 처리 분야에서는 이러한 방법이 저 성능의 주요 원인이

될 수 있는 문제점이 있다.

**발명이 이루고자 하는 기술적 과제**

<13> 본 발명은 상기와 같은 문제점을 해결하기 위해 창작된 것으로서, 본 발명의 목적은 반도체 집적회로에서 긴 비트의 두 숫자의 크기를 비교할 때 뺄셈기에 의한 연산 결과를 통해 판단하지 않고 두 숫자의 각 비트를 MSB로부터 비트단위로 각각 비교하여 먼저 1이 나오는 쪽의 숫자를 큰 수로 결정함으로써 트랜지스터 개수를 줄여 효율적인 크기 비교연산을 수행할 수 있도록 하는 반도체 집적회로의 두 숫자 비교기를 제공함에 있다.

**발명의 구성 및 작용**

<14> 상기와 같은 목적을 실현하기 위한 본 발명은 어느 한 숫자의 각 비트에 대응되는 다른 한 숫자의 각 비트에 대해 우위인 비트를 변환하여 출력하는 슈퍼리어 엔코딩부와, 비교하기 위한 두 숫자에 대해 각각 슈퍼리어 엔코딩부에서 변환되어 출력된 두 우위값의 각 비트를 최상위비트로부터 순차적으로 각 비트를 비교하여 각 비트의 크기를 결정하여 출력하는 비트와이즈 컴페티션부와, 비트와이즈 컴페티션부에서 순차적으로 각 비트의 크기가 결정되어 출력되는 값을 입력받아 크기가 결정된 최초의 상위 비트에 따라 두 숫자의 크기를 판별하는 판별부를 포함하여 이루어진 것을 특징으로 한다.

<15> 본 발명에서 슈퍼리어 엔코딩부는 어느 한 숫자를 반전하여 다른 한 숫자와 노어게이팅하는 것을 특징으로 한다.

<16> 본 발명에서 슈퍼리어 엔코딩부는 두 비트 중 우위인 비트는 '1'로 변환되고 동일한 비트는 '0'으로 변환되는 것을 특징으로 한다.

<17> 본 발명에서 비트와이즈 컴페티션부에서 두 숫자의 최상위 비트는 부호비트가 아닌 것을 특징으로 한다.

<18> 본 발명에서 비트와이즈 컴페티션부에서 최상위 비트가 부호비트인 숫자체계에서는 두 숫자의 최상위 비트를 맞바꾸어 판단하는 것을 특징으로 한다.

<19> 본 발명에서 판별부는 크기가 결정된 최초의 상위 비트 값을 고정하여 출력하는 것을 특징으로 한다.

<20> 이와 같이 이루어진 본 발명은 반도체 집적회로에서 크기를 비교하기 위한 두 숫자의 각 비트에 대해 슈퍼리어 엔코딩부를 통해 우위비트를 변환하여 최상위 비트로부터 각 비트의 크기를 결정하여 최초의 상위 비트값이 크기에 따라 두 숫자의 크기를 판단함으로써 적은 소자의 수로 효율적인 연산을 수행할 수 있도록 한다.

<21> 이하, 본 발명의 바람직한 실시예를 첨부된 도면을 참조하여 설명하며 종래 구성과 동일한 부분은 동일한 부호 및 명칭을 사용한다. 또한 본 실시예는 본 발명의 권리범위를 한정하는 것은 아니고, 단지 예시로 제시된 것이며 당 분야에서 통상의 지식을 가진 자라면 본 발명의 기술적 사상 내에서 많은 변형이 가능할 것이다.

<22> 도 1은 본 발명에 의한 반도체 집적회로의 두 숫자 비교기를 나타낸 블록구성도이다.

<23> 여기에 도시된 바와 같이 어느 한 숫자의 각 비트에 대응되는 다른 한 숫자의 각 비트에 대해 우위인 비트를 변환하여 출력하는 슈퍼리어 엔코딩(superior encoding)부(10)와, 비교하기 위한 두 숫자에 대해 각각 슈퍼리어 엔코딩부(10)에서 변환되어 출력된 두 우위값의 각 비트를 최상위 비트로부터 순차적으로 각 비트를 비교하여 각 비트의 크기를 결정하여 출력하는 비트와이즈 컴페티션(bitwise competition)부(20)와, 비트와이즈 컴페티션부(20)에서 순차적으로 각 비트의 크기가 결정되어 출력되는 값을 입력받아 크기가 결정된 최초의 상위 비트에 따라 두 숫자의 크기를 판별하는 판별부(30)로 이루어진다.

<24> 위와 같이 이루어진 본 발명은 비트 단위로 크기를 비교하기 위해 먼저, 슈퍼리어 엔코딩부(10)를 통해 비교하기 위한 두 숫자의 각 비트에 우위인 비트를 변환하게 된다.

<25> 슈퍼리어 엔코딩부(10)는 한 입력이 다른 입력보다 확실하게 큰 경우에만 '1'로 코딩이 된다. 즉,  $A_{sup}B$ (A superior to B)는 A가 '1'이고 B가 '0'인 경우에만 '1'이 된다. 마찬가지로  $B_{sup}A$ (B superior to B)는 B가 '1'이고 A가 '0'인 경우에만 '1'이 된다. A와 B가 같은 경우는  $A_{sup}B$ ,  $B_{sup}A$  모두 0이 되게 된다.

<26> 도 2는 본 발명에 의한 슈퍼리어 엔코딩부를 나타낸 도면이다.

<27> 여기에 도시된 바와 같이  $A_{sup}B = \sim A \cdot B$ 로,  $B_{sup}A = \sim B \cdot A$ 로 간단하게 인코딩을 표현식으로 나타낼 수 있다.

- <28> 즉, A 숫자의 각 비트에 대한 상위 비트를 확인하기 위해서는 A를 반전한 후 B와 노어게이팅함으로써 구하게 된다.
- <29> 따라서 각 비트 당 NOR gate 하나를 이용하여 간단히 하드웨어로 구현할 수 있게 된다.
- <30> 이와 같이 비교하기 위한 두 숫자를 각 비트별로 슈퍼리어 엔코딩부(10)를 거쳐 출력되는 supA와 supB 값은 비트와이즈 컴페티션부(20)에서 각 비트의 크기를 결정하게 된다.
- <31> 즉, 각 비트별로 슈퍼리어 엔코딩부(10)를 거쳐 변환된 신호인 AsupB<i>와 BsupA<i>는 동시에 1이 될 수 없기 때문에 AsupB와 BsupA 중에서 최상위 비트로부터 '1'이 먼저 나오는 쪽이 더 큰 숫자라고 판단할 수 있게 된다.
- <32> 이와 같은 방법은 최상위 비트가 부호비트가 아닌 숫자 체계(Undsigned number system)에서 판단 가능한 방법이며, 최상위 비트가 부호비트인 숫자 체계(Signed number system)에 적용하기 위해서는 최상위 비트를 서로 맞바꾸어 주어야 한다.
- <33> 이는 최상위 비트가 부호비트인 숫자 체계에서는 최상위 비트는 (-)값을 의미하기 때문에 한 숫자의 최상위 비트가 1이고 다른 숫자의 최상위 비트가 0일 경우, 최상위 비트가 부호비트가 아닌 숫자 체계와는 반대로 최상위 비트가 1인 경우가 반드시 작은 쪽이 되기 때문이다.
- <34> 또한, 최상위 비트가 아닌 비트들은 최상위 비트가 부호비트가 아닌 숫자 체계와 동일하기 때문에, 최상위 비트만 맞바꾸어 주고 나머지 비트들은 같은 방식으로 연결하여 연산하면 최상위 비트가 부호비트인 숫자 체계에서도 올바른 결과를 얻을 수 있다.
- <35> 도 3은 본 발명에 의한 두 숫자 비교 방식을 설명하기 위한 도면이다.
- <36> 여기에 도시된 예는 간단한 4bit의 두 숫자를 비교하는 방식으로써 supA(AsupB)의 경우가 supB(BsupA)의 경우보다 최상위 비트에서부터 '1'이 먼저 나오기 때문에 뒤의 '1'과는 상관없이 A가 B보다 크다고 결정할 수 있다.
- <37> 즉, 슈퍼리어 엔코딩부(10)로부터 대응되는 비트가 동시에 1이 되는 경우는 발생하지 않기 때문에, 최상위 비트로부터 순차적으로 어느 쪽이 먼저 '1'이 발생하는지만 알면 두 숫자의 크기를 비교할 수 있게 된다.
- <38> 도 4는 본 발명에 의한 비트와이즈 컴페티션부를 나타낸 회로구성도이다.
- <39> 여기에 도시된 바와 같이 비트와이즈 컴페티션부(20)는 두 개의 병렬 라인이 있고 각 라인에 비트마다 두 개의 NMOS 트랜지스터가 직렬로 그라운드와 연결되어 있다. 이때 두 라인의 이름은 Aline과 Bline이며, 라인측에 연결된 NMOS 트랜지스터에는 슈퍼리어 엔코딩부(10)를 거친 AsupB와 BsupA 신호가 입력되고, 그라운드측에 연결된 NMOS 트랜지스터에는 START1 ~ START8 신호가 최상위 비트(MSB)쪽에서부터 최하위 비트(LSB)쪽으로 입력된다.
- <40> 따라서 START 신호가 '0' 일 때는 Aline, Bline 모두 '1'로 프리차지 된다. 즉, 그라운드측에 연결된 NMOS 트랜지스터가 풋(foot) 트랜지스터의 역할을 하기 때문에 두 라인의 전압은 '1'로 유지된다. 이후 슈퍼리어 엔코딩부(10)에서 변환이 종료된 후 변환된 값이 입력되면, 풋 트랜지스터의 입력인 START 신호들이 트리거링 되기 시작하게 된다.
- <41> 그러면, 최상위 비트인 START1부터 '0'에서 '1'로 활성화되기 시작하여 START8까지 차례대로 활성화되어 판별부(30)로 출력될 때 각각 인버터를 매개로 출력된다. START 신호 활성화에 의해 해당 비트의 SupA와 SupB 신호는 라인에 영향을 주게 되어 SupA나 SupB 신호가 '1'일 경우에는 라인의 전압을 '0'으로 떨어뜨리게 된다.
- <42> 따라서, START 신호는 최상위 비트로부터 활성화가 되기 때문에 Aline, Bline 중 '0'으로 먼저 떨어지는 쪽이, 슈퍼리어 엔코딩부(10)에서 변환된 신호가 최상위 비트 쪽으로부터 먼저 '1'이 나오는 쪽이 된다. 즉, 최상위 비트 방향에서부터 일정 시간 간격으로 각각의 비트를 경쟁시켜서, 먼저 '1'이 나오는 쪽이 해당 라인을 먼저 '0'으로 떨어뜨리게 되며, 먼저 떨어지는 쪽이 큰 수가 된다.
- <43> 따라서 판별부(30)에서 두 라인 중 먼저 '0'으로 떨어지는 쪽을 감지해 그 결과를 다음 비트의 결과에 상관없이 유지하도록 함으로써 판별 결과를 출력하게 된다.
- <44> 도 5는 본 발명에 의한 판별부를 나타낸 회로구성도이다.
- <45> 본 발명에 의한 판별부(30)는 센스앰프에 피드백 패스가 더해진 형태으로써 START 신호는 센스앰프의 인에이블(enable) 신호와 같은 역할을 한다.
- <46> 이 START 신호가 '0' 일 때는 두 출력 Awin, Bwin을 비롯한 각 노드들이 VDD로 프리차지가 되게 되고, 이 START

신호가 '1'이 되면 회로 동작이 시작되게 된다.

- <47> 판별부(30)의 두 입력은 비트와이즈 컴페티션부(20)의 Aline과 Bline이 각각 인버터를 거친 입력됨에 따라 비트와이즈 컴페티션부(20)에서 비트간 경쟁하여 먼저 이겨 '0'으로 떨어지는 라인은 인버터를 거치면서 먼저 '0'에서 '1'로 올라가게 되고, 센스앰프를 동작시켜 '1'을 유지하고 있던 두 출력 중 상대 라인에 해당하는 출력을 '0'으로 끌어 내리게 된다. 즉, 비트와이즈 컴페티션부(20)에서 A가 이겨 Aline이 Bline보다 먼저 '0'으로 떨어졌다면, 판별부(30)의 출력 Awin과 Bwin 중 Awin은 그대로 '1'이 유지되고 Bwin은 '0'으로 떨어지게 된다.
- <48> 이와 같이 Awin과 Bwin이 결정되어 둘 중 하나가 '0'으로 떨어지게 되면 그 결과가 피드백되어 풋 트랜지스터를 끊게 되고, 결과값은 다음 비트 입력값에 상관없이 초기 결정값이 유지되게 된다.
- <49> 위와 같은 방법으로 판별부(30)는 두 라인 중 먼저 '0'으로 떨어지는 쪽을 감지해 그 결과를 유지하는 역할을 수행하며, 이로써 두 숫자 중 어느 쪽이 큰 쪽인지 알 수 있게 된다.
- <50> 만약 두 수가 같다면, Aline과 Bline은 둘 다 '0'으로 떨어지지 않으므로 판별부(30)의 두 결과는 모두 '1'로 남아있게 된다.
- <51> 따라서 판별부(30)의 두 결과가 모두 '1'로 남아있다면, 두 숫자의 크기가 같다고 판단할 수 있다.
- <52> 본 실시예에서는 8비트 비트와이즈 컴페티션부(20)를 이용하여 두 8비트 수를 비교할 수 있음을 보았으며, 이것은 16비트, 32비트 또는 64비트로 확장되는 것이 가능하다. 하지만, 비트수가 길어지면 트리거링 하는 START신호를 발생시키는데 딜레이가 크게 소요되므로, 한 라인 쌍에 모든 비트를 경쟁시키는 것은 효과적이지 못하다.
- <53> 따라서 32비트처럼 긴 비트 수를 비교할 때에는 8비트 단위로 나누어 8비트 비트와이즈 컴페티션부(20) 4개를 동시에 실행시키고, 마지막에 이것들 중 옳은 결과를 선택해 주는 간단한 믹스(Mux)를 추가시킴으로써 간단하게 구성할 수 있다.
- <54> 즉, 4개의 결과 중 최종 결과를 선택할 때는 최상위 비트로부터 우선순위를 주어 첫 번째 결과의 최상위 비트의 결과를 택하고, 만약 이것이 비겼다면 두 번째 최상위 비트의 결과를 택한다. 마찬가지로 방법으로, 이것이 또 비겼다면 또 다음 최상위 비트의 결과를 택하는 방법으로 비교하면 숫자의 비트 수가 길다고 할지라도 늘어나는 딜레이의 페널티는 크지 않게 된다.
- <55> 위와 같이 비트와이즈 컴페티션을 통해 두 긴 숫자를 비교하는 비교기의 경우 각 비트 당 슈퍼리어 엔코딩을 위해 게이트 2개, 비트와이즈 컴페티션부(20)의 라인에 연결되는 4개의 트랜지스터로 구성할 수 있으며, 이외에 판별부(30)와 믹스만으로 두 긴 숫자의 비교기 구성이 가능함에 따라 기존의 뺄셈기를 기본으로 한 비교기가 소모하는 평균적인 트랜지스터의 개수와 비교하면 1/4 이하임을 알 수 있다.
- <56> 더구나 비트 수가 길어질수록 그 차이는 더욱 크게 되는데 비트와이즈 컴페티션에 의한 본 발명의 비교기는 성능면에서 캐리 carry lookahead 방법을 사용한 뺄셈기보다 조금 느리기는 하지만 큰 차이가 없다.

**표 1**

	Worst case delay	The number of transistors
기존 방식	750 ps	2400 ( Approx. )
본 발명	950 ps	536

<58> 즉, 표 1에 나타낸 바와 같이 32비트의 두 수를 비교할 경우, 트랜지스터의 숫자(면적 관련)와 최대 딜레이의 비교할 때 본 발명에 의해 비트와이즈 컴페티션에 의한 비교기는 훨씬 적은 트랜지스터 개수로도 비슷한 성능으로 두 수의 비교를 효율적으로 수행하게 된다.

**발명의 효과**

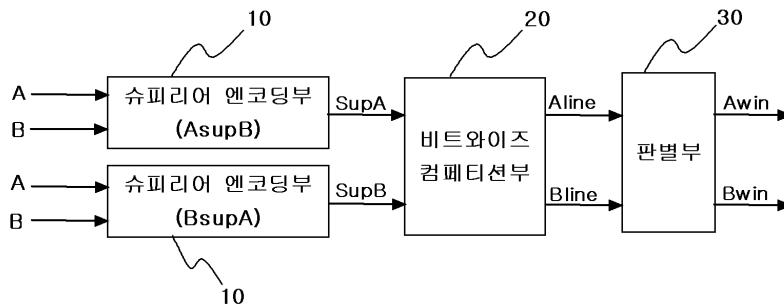
<59> 상술한 바와 같이 본 발명은 반도체 집적회로에서 긴 비트의 두 숫자의 크기를 비교할 때 뺄셈기에 의한 연산 결과를 통해 판단하지 않고 두 숫자의 각 비트를 MSB로부터 비트단위로 각각 비교하여 먼저 1이 나오는 쪽의 숫자를 큰 수로 결정함으로써 트랜지스터 개수를 줄여 효율적인 크기 비교연산을 수행할 수 있는 이점이 있다.

**도면의 간단한 설명**

- <1> 도 1은 본 발명에 의한 반도체 집적회로의 두 숫자 비교기를 나타낸 블록구성도이다.
- <2> 도 2는 본 발명에 의한 슈퍼리어 엔코딩부를 나타낸 도면이다.
- <3> 도 3은 본 발명에 의한 두 숫자 비교 방식을 설명하기 위한 도면이다.
- <4> 도 4는 본 발명에 의한 비트와이즈 컴페티션부를 나타낸 회로구성도이다.
- <5> 도 5는 본 발명에 의한 판별부를 나타낸 회로구성도이다.
- <6> - 도면의 주요부분에 대한 부호의 설명 -
- <7> 10 : 슈퍼리어 엔코딩부
- <8> 20 : 비트와이즈 컴페티션부
- <9> 30 : 판별부

도면

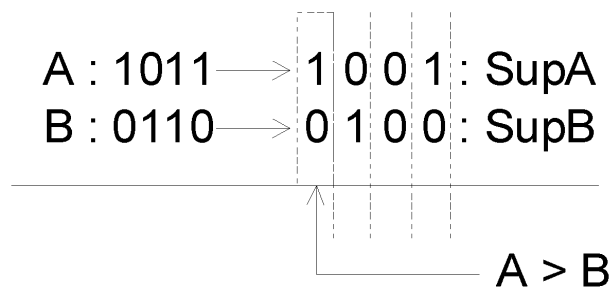
도면1



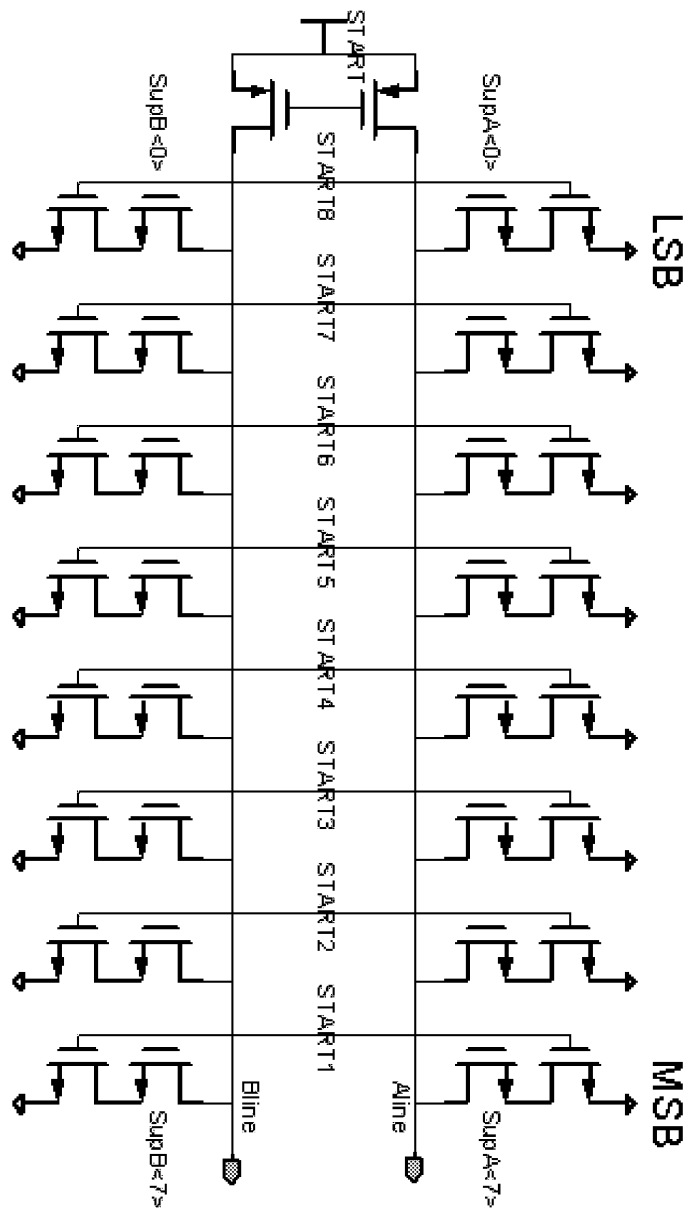
도면2



도면3



도면4



도면5

