(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2012년11월02일
(11) 등록번호 10-1196048
(24) 등록일자 2012년10월24일

(51) 국제특허분류(Int. Cl.)
G06F 13/16 (2006.01) *G06F 13/14* (2006.01)
G06F 13/00 (2006.01) *G06F 9/46* (2006.01)
(21) 출원번호 10-2006-7010659
(22) 출원일자(국제) 2004년10월27일
심사청구일자 2009년10월27일
(85) 번역문제출일자 2006년05월30일
(65) 공개번호 10-2006-0111544
(43) 공개일자 2006년10월27일
(86) 국제출원번호 PCT/US2004/035863
(87) 국제공개번호 WO 2005/045727
국제공개일자 2005년05월19일
(30) 우선권주장
10/698,905 2003년10월31일 미국(US)
(56) 선행기술조사문헌
US06105094 A1*
US20020138687 A1*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
소닉스, 인코퍼레이티드
미국 캘리포니아 94040 마운틴 뷰 수트 620 웨스트 엘 카미노 리얼 2440
(72) 발명자
베버, 볼프-디트리히
미국 95120 캘리포니아 샌어제이 민더 드라이브 5851
(74) 대리인
남상선

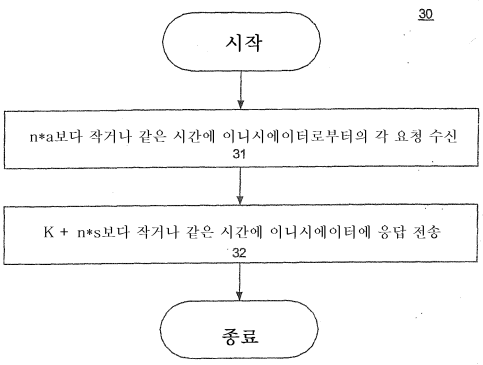
전체 청구항 수 : 총 14 항 심사관 : 엄인권

(54) 발명의 명칭 **다수의 프로세서 간 메모리 액세스의 스케줄링**

(57) 요 약

일반적으로, 서비스 품질(QoS) 모델을 구현하기 위한 방법 및 장치가 개시된다. 이니시에이팅(initiating) 네트워크 장치와의 서비스 품질(QoS) 계약이 충족될 수 있다. {서수 × 도착 간격}보다 작거나 같은 제 1 시간에 이니시에이팅 네트워크 장치 이니시에이터로부터 요청이 수신될 수 있다. 서수는 요청들의 그룹 사이에서 요청의 위치를 나타낸다. 서비스되었던 요청은 {상수항 + 서수 × 서비스 간격}보다 작거나 같은 제 2 시간에 이니시에이터로 반환될 수 있다.

대 표 도 - 도3



특허청구의 범위

청구항 1

집적회로에서 이니시에이터(initiator)(11), 상호접속부(13) 및 타깃(12) 간의 서비스 품질(QoS) 계약을 충족(satisfy)시키기 위한 방법으로서,

데드라인들(deadlines)을 특정하는 도착 모델(40)에 따라 중재 포인트(arbitration point)(23)에서 요청들의 그룹을 수신하는 단계 - 상기 데드라인들에서 또는 상기 데드라인들 이전에, 상기 요청들(42a-42g)의 그룹 내의 각 요청이 수신되어야 하고, 상기 도착 모델에 따른 상기 데드라인들은 {서수(ordinal number) × 도착 간격(43)}과 같으며, 상기 서수는 상기 요청들(42a-42g)의 그룹 사이에서 상기 요청의 위치를 나타냄 -;

상기 요청들(42a-42g)의 그룹에 대한 응답들이 데드라인들을 특정하는 서비스 모델(70)에 따라 상기 이니시에이터(11)에게 다시 서비스되는 시기를 상기 타깃(12)에 의해 결정하는 단계 - 상기 데드라인들에서 또는 상기 데드라인들 이전에, 상기 요청들(42a-42g)의 그룹 내의 각 요청이 서비스되어야 하고, 상기 서비스 모델에 따른 상기 데드라인들은 {상수항 + 상기 서수 × 서비스 간격}과 같음 -; 및

상기 QoS 계약을 충족시키기 위해 상기 도착 모델 및 상기 서비스 모델이 충족되었는지 여부를 결정하는 단계 - 상기 QoS 계약은 충족되는 상기 도착 모델(40) 및 상기 서비스 모델(70) 둘 다를 기반으로 함 -

를 포함하는 서비스 품질(QoS) 계약을 충족시키기 위한 방법.

청구항 2

제 1 항에 있어서,

칩 상에 모두 위치되는 상기 이니시에이터, 타깃, 상기 상호접속부, 및 QoS 유닛을 이용하여 상기 QoS 계약을 충족시키는 단계를 더 포함하는,

서비스 품질(QoS) 계약을 충족시키기 위한 방법.

청구항 3

제 1 항에 있어서,

상기 서비스 간격은 상기 도착 간격보다 큰,

서비스 품질(QoS) 계약을 충족시키기 위한 방법.

청구항 4

제 1 항에 있어서,

상기 서비스 간격은 상기 도착 간격과 같은,

서비스 품질(QoS) 계약을 충족시키기 위한 방법.

청구항 5

제 1 항에 있어서,

상기 요청들의 그룹에 대해 구체적으로 상기 상수항을 결정하는 단계를 더 포함하는,

서비스 품질(QoS) 계약을 충족시키기 위한 방법.

청구항 6

제 5 항에 있어서,

상기 도착 간격은 요청된 도착들에 대해 미리 결정된 허용 간격(accepted interval)인,

서비스 품질(QoS) 계약을 충족시키기 위한 방법.

청구항 7

제 1 항에 있어서,

상기 도착 모델을 충족시키기 위해 상기 {서수×도착 간격} 보다 작거나 같은 제 1 시간에서 상기 집적회로의 상기 이니시에이터로부터 요청을 수신하는 단계;

상기 서비스 모델을 충족시키기 위해 상기 {상수항 + 상기 서수 × 서비스 간격} 보다 작거나 같은 제 2 시간에서 상기 타깃에 의해 상기 이니시에이터에게 서비스되었던 상기 이니시에이터로부터의 상기 요청을 반환(return)하는 단계; 및

상기 집적회로의 상기 이니시에이터와 상호접속부 사이의 경계에서 상기 제 1 및 제 2 시간을 측정하는 단계를 더 포함하는,

서비스 품질(QoS) 계약을 충족시키기 위한 방법.

청구항 8

시스템으로서,

이니시에이터(11)와 다수의 타깃(12a-12c) 사이에 연결된 상호접속부(13) - 동일한 이니시에이터로부터의 요청들의 그룹에서의 제 1 요청(42a) 및 추가적인 요청들(42b-42g)이 {서수 × 도착 간격(43)}보다 작거나 같은 제 1 시간량 이전에 도착하는 경우, 상기 이니시에이터(11)에 의해 도착 모델이 충족되고, 상기 서수는 요청들(42a-42g)의 그룹 사이에서 상기 요청의 위치를 나타냄 -; 및

상기 이니시에이터(11)와 상기 상호접속부(13) 사이의 경계(16)에서 측정되는 상기 제 1 시간량에서 상기 도착 모델을 충족시키기 위해, 상기 상호접속부(13)에 명령들을 발행(issue)하기 위한 서비스 품질(QoS) 로직 유닛(27)

을 포함하고, 제1 타깃(12a)은 상기 제 1 요청(42a) 및 상기 추가적인 요청들(42b-42g)이 {상수항 + 상기 서수 × 서비스 간격}보다 작거나 같은 제 2 시간량 이전에 서비스되는 경우 상기 서비스 모델을 충족시키도록 구성되며, 상기 제 2 시간량은 상기 제1 타깃(12a)과 상기 상호접속부(13) 사이의 경계(28)에서 측정되고,

상기 서비스 품질(QoS) 로직 유닛(27), 상기 제1 타깃(12a) 및 상기 상호접속부(13)는 함께 QoS 계약을 충족시키도록 구성되며, 상기 QoS 계약은 충족되는 상기 도착 모델 및 상기 서비스 모델 둘 다를 기반으로 하는,

시스템.

청구항 9

제 8 항에 있어서,

상기 서비스 간격은 상기 도착 간격보다 큰,

시스템.

청구항 10

제 8 항에 있어서,

상기 상수항은 상기 요청들의 그룹에 대해 구체적으로 결정되는,

시스템.

청구항 11

제 10 항에 있어서,

상기 QoS 유닛은 상기 타깃과 상기 상호접속부 사이에 연결되는,

시스템.

청구항 12

제 8 항에 있어서,

상기 QoS 계약은 칩 상에 모두 위치되는 상기 이니시에이터, 상기 타깃, 상기 상호접속부 및 상기 로직 유닛을 이용하여 충족되는,

시스템.

청구항 13

제 12 항에 있어서,

상기 QoS 유닛은 상기 상호접속부의 일부인,

시스템.

청구항 14

제 8 항에 있어서,

상기 QoS 유닛은 상기 타깃의 일부인,

시스템.

청구항 15

삭제

청구항 16

삭제

청구항 17

삭제

청구항 18

삭제

청구항 19

삭제

청구항 20

삭제

청구항 21

삭제

청구항 22

삭제

청구항 23

삭제

청구항 24

삭제

청구항 25

삭제

청구항 26

삭제

청구항 27

삭제

청구항 28

삭제

청구항 29

삭제

청구항 30

삭제

청구항 31

삭제

청구항 32

삭제

명 세 서

기 술 분 야

[0001] 본 발명은 일반적으로 통합 시스템에 관한 것이며, 양상은 구체적으로 통합 시스템에서의 서비스 품질 보증에 관한 것이다.

배 경 기 술

[0002] SoC(System-on-Chip)은 셀룰러폰, 디지털 카메라, 셋탑박스(STB) 등의 시스템에 필요한 전자 회로 및 부품들의 대부분 또는 전부를 통합하는 집적 회로(IC)이다. SoC는 중앙 처리 유닛(CPU), 직접 메모리 액세스(DMA) 유닛, 메모리, 입력/출력(I/O) 회로, 및 특정 애플리케이션에 필요한 다른 회로들과 같은 개별 칩 상에 따로 있을 수 있는 여러 회로를 통합할 수 있다. 특정 애플리케이션에 필요한 모든 회로를 하나의 IC에 포함함으로써, 시스템 제조 비용 및 시스템 크기가 감소할 수 있고, 시스템의 신뢰도가 개선될 수 있다.

[0003] SoC는 상호 작용하는 여러 엔티티를 포함하는 단일 칩이다. 엔티티들은 일반적으로 SoC의 제조업자에 의해 생산되기보다는 다른 판매업자로부터 라이센싱되기 때문에 지적 재산(IP: Intellectual Property) 코어라 할 수도 있다. CPU와 같은 이니시에이터(initiator)는 메모리와 같은 타깃에 서비스 요청을 발행한다. 예를 들어, CPU는 메모리의 일부에 대한 액세스를 필요로 할 수도 있다. CPU는 메모리에 특정 데이터 요청을 발행하게 된다. 그리고 메모리는 그 요청을 서비스하여 요청한 데이터를 CPU에 돌려보내게 된다. 이니시에이터 및 타깃은 상호접속부를 통해 접속된다.

[0004] 서비스 품질(QoS)은 요청이 얼마나 신속하게 충족되는지에 관련한 성능의 기대라 할 수도 있다. 예를 들어, 이니시에이터는 요청을 발행할 수 있고, 이들 요청이 타깃에 의해 특정 시간 내에 충족될 것을 예상할 수 있다. 성능은 여러 다른 방법으로 지정될 수 있다. 대역폭 성능은 단위 시간당 특정 개수의 요청 수신에 관련된다. 지연 성능은 특정 요청이 반환되는 시간에 관련된다. 지터 성능은 요청들 또는 도착하는 응답들 간의 시간 편차에 관련된다.

[0005] SoC에 사용되는 이니시에이터들은 통상적으로 매우 까다로운 서비스 요건을 갖기 때문에 SoC를 이용할 때 QoS 표준이 특히 중요하다. 예를 들어, (CPU와 같은) 어떤 이니시에이터들은 까다로운 지연 요건을 가지며, 신속히 충족될 것을 요구한다. (통신 인터페이스와 같은) 다른 이니시에이터들은 대역폭 및 지터 성능에 더 민감하다.

어떤 SoC들은 SoC에 확실한 성능 보증이 충족될 것을 보장할 수 없는 QoS 모델로부터 피해를 받는다.

발명의 상세한 설명

[0006] 서비스 품질(QoS) 모델에 관한 방법 및 장치가 개시된다. QoS 모델에 따르면, {서수(ordinal number) × 도착 간격}보다 작은 시간과 같거나 더 작은 제 1 시간에 이니시에이터로부터 요청이 수신되며, 서수는 요청 그룹 사이에서 요청의 위치를 나타낸다. 또한, 상기 모델에 따르면, 서비스된 요청이 {상수항 + 서수 × 서비스 간격}과 같거나 작은 제 2 시간에 이니시에이터로 반환된다.

실 시 예

[0015] 다음 설명에서는, 본 발명의 전반적인 이해를 제공하기 위해, 지정된 구성요소, 접속, 그룹의 요청 회수 등의 예와 같이 많은 특정 항목이 설명된다. 그러나 본 발명은 이러한 특정 항목 없이 실시될 수도 있다는 점이 당업자들에게 명백할 것이다. 다른 경우에는, 본 발명을 쓸데없이 불명료하게 하지 않도록 잘 알려진 구성요소들이나 방법들은 상세히 설명되지 않고 블록도로서 설명한다. 이와 같이, 설명하는 특정 항목들은 예시일 뿐이다. 특정 항목들은 달라질 수 있으며 여전히 본 발명의 진의 및 범위 내에 있는 것으로 기대할 수 있다.

[0016] 일반적으로, 서비스 품질(QoS)을 구현하는 방법 및 장치가 개시된다. 본 발명의 실시예에 따르면, SoC(System-on-Chip) 또는 이니시에이터와 타깃 사이의 서비스를 보증할 필요가 있는 다른 시스템상에서 QoS 모델이 구현될 수 있다. 이니시에이터 및 타깃은 상호접속부를 통해 접속된다. 중앙 처리 유닛(CPU)과 같은 이니시에이터는 랜덤 액세스 메모리(RAM)와 같은 타깃으로부터의 서비스를 요청할 수 있다. 스레드는 이니시에이터에서 타깃으로 요청을 전송하는 채널이다. QoS 유닛에 의해 안내되는 상호접속부는 어떤 스레드가 타깃에 의해 서비스될지를 결정한다. QoS 모델에 따르면, 이니시에이터와 나머지 시스템 사이에 계약(contract)이 이루어질 수 있다. 계약에 따르면, 이니시에이터는 특정 요청이 특정 데드라인들까지 도착될 것을 보장하고, 나머지 시스템은 이들 요청이 다른 데드라인까지 서비스될 것을 보장한다. QoS 모델은 도착 모델 및 서비스 모델을 포함한다. 도착 모델은 이니시에이터로부터 요청을 수신하는 데드라인을 기술하고, 서비스 모델은 타깃 및 상호접속부로부터 서비스를 수신하는 데드라인을 기술한다. 도착 모델에 따르면, 이니시에이터로부터의 요청은 n*a 시간 전에 도착해야 하며, 여기서 'n'은 제 1 요청 후의 요청 회수이고, 'a'는 미리 결정된 도착 간격이다. 서비스 모델에 따르면, K + n*s보다 작은 시간에 응답이 발행되며, 여기서 'K'는 미리 결정된 상수항이고, 's'는 미리 결정된 서비스 간격이다.

[0017] 다른 실시예에 따르면, 특정 스레드 동안 할당 카운트가 유지된다. 할당 카운트는 특정 스레드가 서비스되고 있는지 여부를 추적할 수 있다. 할당 카운트는 일정한 간격으로 증분(increment)할 수 있으며, 스레드가 서비스될 때 감소한다. 할당 카운트는 어떤 스레드가 서비스될지를 결정하는데 사용될 수 있다. 그 결정은 할당 카운트가 현재 양(+)인지 여부에 기초할 수 있다. [DEW1] 할당 카운트는 아이들(idle) 스레드에 너무 높은 우선순위가 부여되지 않음을 보장하도록 양의 한계를 가질 수 있다. 양의 한계는 더 높은 우선순위 요구가 서비스되고 있기 때문에 서비스되지 않는 더 낮은 우선순위의 대역폭 할당 스레드가 언젠가는 그 할당을 수신할 것을 보증하도록 조정될 수도 있다.

[0018] 도 1은 본 발명의 일 실시예에 따른 예시적인 SoC를 나타낸다. SoC(10)는 상호접속부(13)를 통해 여러 타깃(12)에 연결된 여러 이니시에이터(11)를 포함한다. 여러 이니시에이터(11a, 11b, 11c)가 도시된다. 이니시에이터(11)의 단순한 참조 부호는 임의의 이니시에이터(11a~11c)가 그 경우에 적용될 수 있음을 지시한다. 타깃(12a~12c) 등에도 동일하게 들어맞는다. 이니시에이터(11)는 CPU, 직접 메모리 액세스 유닛(DMA), 그래픽 시스템, 오디오 시스템 등의 장치를 포함할 수 있다. 타깃들은 캐시 메모리, 랜덤 액세스 메모리(RAM), 판독 전용 메모리(ROM), 주변 장치, DMA 유닛, 레지스터 프로그래밍 인터페이스 등의 장치를 포함할 수 있다. 이니시에이터(11)는 일반적으로 타깃(12)에 의한 서비스를 요청한다. 상호접속부(13)는 다양한 이니시에이터(11)를 다양한 타깃(12)에 연결한다. 일반적으로, 어떤 이니시에이터(11)라도 임의의 타깃(12)으로부터의 서비스를 요청할 수 있다.

[0019] 이니시에이터(11)는 상호접속부(13)에 요청(14)을 발행하고 상호접속부(13)로부터 응답(15)을 수신함으로써 타깃(12)과 통신한다. 요청(14)은 예를 들어 메모리 판독 요청일 수도 있다. 대응하는 응답(15)은 그 요청을 충족시키는 데이터가 된다. 요청한 이니시에이터(11)에 응답을 전송함으로써, 요청받은 타깃(12)은 요청을 "서비스"했다고 한다. QoS 모델은 특정 이니시에이터(11)와 상호접속부(13)과의 경계(16)에서 이니시에이터(11)와 나머지 시스템과의 계약을 기술한다. QoS 모델은 이니시에이터(11)로부터의 요청을 수신하는 데드라인 및 이니

시에이터(11)에 응답을 반환하는 데드라인을 지정한다.

[0020] 도 2는 본 발명의 일 실시예에 따른 QoS 모델을 구현하는 시스템을 나타낸다. 시스템(20)은 SoC 또는 QoS 관리를 필요로 하는 다른 시스템일 수도 있다. 여러 이니시에이터(11a~11c)가 타깃(12a)과 통신한다. 여기서는 타깃(12a)이 지정되지만, 임의의 타깃(12)이 사용될 수 있는 것으로 이해한다. 이니시에이터(11a~11c)는 타깃(12a)에 의해 서비스되어야 하는 요청들의 그룹을 발행한다. 각 이니시에이터(11a~11c)는 하나 이상의 스레드(21a~21c) 상에 요청을 발행한다. 스레드(21)는 물리 채널(22)에 설정되는 가상 채널이다. 여기 도시한 바와 같이, 각 이니시에이터(11a~11c)는 그 이니시에이터(11a~11c)에 대응하는 각자의 전용 물리 채널(22a~22c)을 갖고 있다. 여러 스레드가 동일한 물리 채널(22) 상에 멀티플렉싱될 수 있다. 다른 스레드로부터의 요청이 상호접속부(13) 내부의 하나 이상의 중재 포인트(23)에서 수신된다. 중재 포인트(23)는 타깃(12a)에 지정될 수 있다. 다른 실시예에서, 중재 포인트(23)는 여러 다른 타깃(12)을 서비스할 수도 있다. 중재 포인트(23)는 언제, 그리고 어떤 순서로 타깃(12a)에 요청이 제시되는지를 결정하고, 타깃(12a)은 그 서비스 타이밍, 다중 스레드의 경우에는 일부 서비스 순서를 결정한다. [DEW2] 요청이 서비스될 때, 요청은 응답 스레드(24a~24c)로서 리턴 채널(25a~25c)을 따라 이니시에이터(11)에 반환된다. 응답은 타깃(12a)에 의해 서비스되었고, 분할 포인트(26)에 의해 원래의 이니시에이터(11)로 되돌아간다.

[0021] 물리 채널(22, 25)에는 여러 개의 스레드가 도시된다. 스레드(21a~21c)는 이니시에이터(11a~11c)로부터의 요청을 운반하는 가상 채널인 이니시에이터(11a~11c)로부터의 요청 스레드이다. 이들은 물리 채널(22)의 일부에만 도시하였지만, 스레드(21)는 물리 채널(22)의 길이를 "가상으로" 연장한다. 마찬가지로, 응답 스레드(24)는 물리적 응답 채널(25)을 따르는 가상 채널이다. 요청 채널(26)은 상호접속부(13)로부터 타깃(12a)으로 요청을 전송한다. 요청 채널(26)은 QoS 유닛(27)에 의해서도 인지된다. QoS 유닛(27)은 채널(28a)을 통해 요청 채널에, 그리고 채널(28b)을 통해 중재 포인트(23)에 연결된다. 응답 채널(29)은 QoS 유닛으로부터 상호접속부(13)로 응답을 전송한다. 알 수 있듯이, 스레드(21a, 21b, 21c)는 요청 채널(26) 상에서 멀티플렉싱된다. 마찬가지로, 스레드(24a, 24b, 24c)는 응답 채널(29) 상에서 멀티플렉싱된다.

[0022] QoS 유닛(27)은 채널(28b)을 통해 상호접속부(13)에 명령을 발행한다. QoS 유닛(27)은 타깃(12a) 내부나 상호접속부(13) 내부에 있을 수도 있고, 또는 도시한 바와 같이 개별적일 수도 있다. 상호접속부(13)가 궁극적으로 타깃(12a)에 어떤 요청이 발행되는지를 결정하지만, QoS 유닛(27)은 계약 및 QoS 모델에 따라 상호접속부(13)를 안내한다. 각 스레드(21)에 대해, QoS 유닛(27)은 다른 이니시에이터(11)와의 각종 계약에 따른 서비스를 위해 타깃(12a)에 언제 스레드(21)가 제시될지를 결정한다. QoS 유닛(27), 타깃(12a) 및 상호접속부(13)가 함께 QoS 계약의 요건을 충족시킨다. 이들 계약의 특징은 하기에 설명한다.

[0023] 도 3은 본 발명의 일 실시예에 따른 QoS 모델을 설명하는 흐름도이다. 일 실시예에서, QoS 유닛(27), 상호접속부(13) 및 타깃(12)이 프로세스(30)를 구현한다. 블록(31)은 도착 모델을 설명한다. 블록(31)에서는 이니시에이터로부터 요청 그룹이 수신된다. QoS 모델에 따라 각 요청은 n*a보다 작거나 같은 시간에 수신되며, n은 제 1 요청 후의 요청 회수를 기술하는 서수이다. 예를 들어, n은 제 1 요청에 대해 0, 제 2 요청에 대해 1이 되는 식이다. 변수(a)는 도착 간격과 관련되며, 이는 요청 도착에 대한 시스템의 허용 간격과 관련된 미리 결정된 시간이다. 이 모델에 따르면, 7개의 요청으로 이루어진 그룹이 제 1 요청이 도착한 후 시간(6a) 전에 도착한다. 그룹에서 각각의 요청은 제 1 요청 후 시간(n*a)에 도착할 수 있지만, 요청은 그 전에 언제든지 도착할 수도 있다. 예를 들어, 제 1 요청 후 a 전에 언제든지, 제 2 요청 후 2a 전에 언제든지 도착할 수 있는 식이다. 모델은 각 요청이 도착해야 하는 또는 도착하기 전의 데드라인을 설정한다. 도착 간격(a)은 애플리케이션에 따라 시스템에 의해 고정될 수도 있고 가변할 수도 있다. 제 1 로직이 도착 모델이 충족되는지 여부를 결정할 수 있는 것으로 이해한다.

[0024] 블록(32)은 서비스 모델을 설명한다. 블록(32)에서는 K + n*s보다 작거나 같은 시간에 이니시에이터에 응답이 전송된다. K 항은 이러한 양을 서비스의 지연 및 지터로서 커버하는 상수항이다. K 항은 각 그룹에 더해지며, 타깃 서비스 그룹에 더 큰 허용 한계를 제공하여 더 높은 우선순위를 가질 수 있는 다른 그룹들을 스케줄링하거나 전체 시스템 효율을 높일 수 있다. 변수 s는 변수 a와 유사한 서비스 간격이다. K 및 s 항은 특정 시스템에 고정될 수도 있고, 또는 이니시에이터(11), 타깃(12) 등에 따라 변경될 수도 있다. 타깃(12), 상호접속부(13) 및 QoS 유닛(27)을 포함하는 로직은 도착 모델이 충족되었다고 판단되면 서비스 모델을 만족시킬 수 있다.

[0025] QoS 모델은 두 부분: 도착 모델 및 서비스 모델을 포함한다. 도 4a~4c는 본 발명의 일 실시예에 따른 도착 모델을 나타낸다. 도 4a는 도착 모델(40)에 따른 요청 그룹을 나타낸다. 도착 모델(40)은 언제 요청(42)이 수신되는지를 지시하는 타임 라인(41)을 포함한다. 요청 그룹(44)은 여러 개의 개별 요청(42)을 포함한다. 모델

(40)은 계약을 만족시키도록 요청(42)이 수신되어야 하는 일련의 데드라인을 포함한다. 요청 간격(43 또는 a)은 나머지 시스템에 의해 요청이 수신되어야 하는 시간으로 데드라인을 나타낸다. 그룹에서 각각의 요청에 대해, 요청은 다음 식으로 설정되는 시간 전에 도착해야 한다:

n*a

여기서, n은 그룹의 제 1 요청 후의 순차적인 번호에 대응한다(예를 들어, 제 3 요청(42c)에서 n=2가 된다). 제 1 응답의 도착 시간은 시간(0)으로 정해질 수 있다.

[0026] 요청(42a~42g)을 포함하는 요청 그룹(44)은 이니시에이터(11)에 의해 상호접속부(13)에 전송될 수 있다. 이니시에이터(11)는 단일 타깃(12)에 요청 그룹(44)을 발행한다. QoS 계약에 따라, {그룹의 요청 회수 × 요청 간격(a, 43)}보다 작은 시간보다 작거나 같은 시간에 나머지 시스템에 의해 전체 요청 그룹(44)이 수신된다. 각 개별 요청은 {특정 요청 회수보다 작은 수 × a} 전에 수신된다. 예를 들어, 제 2 요청(42b)은 시간(1a) 전에 수신되고, 제 3 요청(42c)은 시간(2a) 전에 수신된다. 알 수 있는 바와 같이, 그룹(44)의 각 개별 요청은 n*a보다 작거나 같은 시간에 도착하며, QoS 도착 모델은 요청 그룹(44)에 대해 충족된다. 또한, 그룹(44)의 각 개별 요청은 그 도착 데드라인(n*a) 직전에 도착한다는 점을 알 수 있다. 이러한 동작은 통신 및 스트리밍 미디어 등의 애플리케이션에서 일반적인 등시성 데이터 생성 프로세스의 특징이다.

[0027] 도 4b는 2개의 요청 그룹을 나타낸다. 요청 그룹(44)이 2개의 요청 그룹(51, 52)으로 분할되었으며, 이는 요청(42d~42g)이 요청(42a~42c)과 그룹화될 때 도착 모델을 만족시키지 않기 때문에 필요할 수도 있다. 요청 그룹(51)은 요청(42a~42)을 포함하고, 요청 그룹(52)은 요청(42d~42g)을 포함한다. 도착 모델을 만족시키기 위해, 요청(42c)은 요청(42a)이 도착한 후 2a 전에 도착해야 한다. 마찬가지로, 도착 모델을 만족시키기 위해, 요청(42g)은 요청(42d)이 도착한 후 2a 전에 도착해야 한다.

[0028] 도 4c는 수신된 요청 그룹을 나타낸다. 요청 그룹(61)은 요청(42a~42g)을 포함한다. QoS 모델에 따르면, 요청(42b)은 시간(a) 전에 도착해야 하고, 요청(42c)은 시간(2a) 전에 도착해야 하는 식이다. 도 4c에서 알 수 있듯이, 요청(42f)은 요청(42f)을 수신하는 데드라인이 시간(5a)이더라도 시간(2a) 전에 수신된다. 모델(60)은 이니시에이터(11)가 요청을 마음대로 일찍 전송한다는 것을 나타낸다. 요청(42g)은 시간(6a) 전에 수신된다. 알 수 있듯이, 제 6 요청(42f) 및 제 7 요청(42g)이 수신되는 시간 사이에 큰 갭이 있다. 그러나 요청(42a~42g)은 모두 QoS 모델에 따라 수신되기 때문에, 이니시에이터(11)는 계약을 만족한다.

[0029] 도 5a 및 5b는 본 발명의 일 실시예에 따른 서비스 모델을 나타낸다. 도 5a에 따르면, 서비스 모델(70)은 요청(42a~42d)을 포함하는 요청 그룹(71)을 포함한다. 도착 모델에 따르면, 그룹(71)은 시간(3a) 전에 수신되어야 한다. 타임 라인(72)은 특정 요청들이 서비스되는 시간을 나타낸다. 서비스 모델에 따르면, 요청 그룹은 다음과 같은 시간 전에 서비스되어야 한다:

K + n*s

여기서, K는 상수항이고 s는 서비스 간격이다. K 항은 계약에 포함된 항이고, 각 요청 그룹에 적용된다. K 항은 타깃(12)에 그룹(71)에 서비스할 여분의 시간을 제공한다. 타깃(12)은 요청 그룹(71)에 서비스할 때 원하는 대로 K 항을 분할할 수 있다. K 항은 초기 지연 항일 수 있으며, 또는 QoS 유닛(27)에 의해 사용될 수 있지만 이것이 바람직하다. K 항은 예를 들어 타깃(12)에 다른 이니시에이터(11)로부터의 요청을 서비스할 더 많은 시간을 제공하는데 사용될 수 있다. 일 실시예에서, 서비스 간격(s)은 도착 간격(a)보다 크거나 같은 시간이다. 요청은 도착할 때까지 서비스될 수 없기 때문에, 서비스 간격(s)은 도착 간격(a)보다 반드시 크거나 같다. a가 s보다 작다면, 서비스 모델은 마치 a가 s와 동일한 것처럼 동작한다. 일 실시예에서, a 및 s는 동일한 것이 바람직하다. 서비스 간격(s)은 공칭(nominal) 대역폭 항으로 생각할 수도 있다. 이 예에서, K 항은 간결성을 위해 서비스 간격과 동일하다. 그러나 K 항은 서비스 간격(s)과 상관없이 선택될 수 있는 것으로 이해한다. 또한, 이 예에서 서비스 항에는 s=1.5a의 임의의 값이 주어진다.

[0030] 서비스 모델에 따르면, 4개의 요청을 포함하는 전체 그룹(71)은 K + n*s와 같은 시간으로 서비스되어야 하며, 이는 s + 3*s = 4s(또는 6a)이다. 이 예에서 K 항에 s의 값이 할당된다는 점을 상기하면, s + 0*s = s이기 때문에 제 1 요청(42a)은 시간(s)으로 서비스되어야 한다. 마찬가지로, s + 1*s = 2s이기 때문에 제 2 요청(42b)은 시간(2s)으로 서비스되어야 한다. 상호접속부(13) 및 타깃(12)은 여기서 K 항을 이용하여 제 1 요청(42a)의 서비스를 지연시킨다. 도 5a에서 알 수 있듯이, 각 요청은 그 각각의 데드라인 전에 수신되기 때문에 그룹(71)은 모델에 따라 서비스되었다.

[0031] 도 5b는 그룹(71)의 다른 서비스를 나타낸다. 이 예(80)에서, K 항에 s의 값이 할당된다는 점을 상기하면, 제

1 및 제 2 요청(42a, 42b)은 시간(s) 전에 서비스된다. 제 3 요청(42c)은 2s에 서비스되고, 제 4 요청(42d)은 4s까지 서비스되지 않는다. 제 4 및 최종 요청은 모델에 따라 4s까지 서비스될 필요 없으며, 타깃(12)이 3개의 제 1 요청(42a~42c)을 일찍 서비스했기 때문에, 타깃(12)은 시간(4s) 전에 최종 요청(42d)이 서비스되는 한 다른 요청들을 마음대로 서비스한다. 도 5b에서 알 수 있듯이, 그룹(71)은 모델에 따라 서비스되었다.

[0032] 도 6은 본 발명의 일 실시예에 따른 할당 카운트를 나타낸다. 각 스레드에 대해 QoS 유닛(27)에 할당 카운트(90)가 유지될 수 있다. 할당 카운트(90)는 일반적으로 특정 스레드가 서비스되고 있는지 여부를 측정한다. 일 실시예에서, 3가지 타입의 스레드: 타깃(12) 대역폭의 미리 할당된 부분 내에 유지되는 한 다른 모든 스레드를 통해 서비스에 대한 우선순위가 부여되는 높은 우선순위의 스레드, 2) 일반적으로 타깃(12) 대역폭의 일부에 확보된 대역폭 할당 스레드, 및 3) 타깃(12)이 서비스하기 위한 여분의 대역폭을 가질 때마다 서비스되는 최선의 스레드가 있다. 할당된 대역폭 및 우선순위 스레드는 할당 카운트(90)를 이용하여 모니터링된다. 할당 미터(91)는 특정 스레드에 발행된 크레디트 수의 예시이다. 할당 미터(91)는 양의 한계(92) 및 음의 한계(93)를 갖고 있다. 이들 한계는 하기에 설명한다.

[0033] 할당 카운트(90)는 대역폭 할당이 이루어지는 스레드(즉, 높은 우선순위 및 대역폭 할당 스레드) 사이의 우선순위를 결정하는데 사용될 수 있다. 일반적으로, 스레드가 서비스되지 않는다면, 흔히 높은 우선순위 스레드가 특정 타깃(1)의 서비스를 요청하기 때문에, 할당 카운트(90)는 점진적으로 양수가 될 것이다. 반대로, 스레드가 할당된 것보다 많은 서비스를 수신했다면, 그 할당 카운트는 음수가 될 것이다. 음의 할당 카운트(90)는 그 스레드의 우선순위를 강등시켜, 다른 스레드에 더 나은 서비스 수신 기회를 제공할 수 있다.

[0034] 규칙적인 간격으로 할당 카운트(90)가 증분된다. 예를 들어, 시간(0)에서 특정 스레드의 할당 카운트(90)는 0이다. 시간(t)에서 스레드에 하나의 크레디트가 발행된다. 따라서 스레드가 서비스를 요청하지 않았다면, 할당 카운트(90)는 +1의 카운트로 양수가 된다. 스레드가 서비스를 수신하면, 크레디트가 데비트(debiting)된다. 예를 들어, 시간(t)에 스레드가 서비스를 한번 요청한다면, 스레드의 할당 카운트(90)는 0이 되는데, 스레드는 시간(t)에 하나의 크레디트(규칙적인 크레디트)를 수신했고, 요청을 충족시킴으로써 그 크레디트를 데비트했기 때문이다. 할당 카운트는 음수가 될 수도 있다. 예를 들어, 시간(t)에 스레드가 이미 서비스를 두 번 요청했다면, 스레드는 하나의 크레디트만을 수신하게 되고, 2개의 크레디트를 데비트하여, -1의 할당 카운트(90)가 된다.

[0035] 할당 카운트(90)는 양의 한계(92) 및 음의 한계(93)를 갖는다. 여기에 나타낸 바와 같이, 양의 한계(92)는 +7의 크레디트이고, 음의 한계(93)는 -7의 크레디트이다. 스레드가 장시간 동안 아이들 상태라면, 스레드는 과도한 크레디트를 누적하게 된다. 그 결과, 스레드가 서비스 요청을 다시 시작하더라도, 할당 카운트(90)는 결코 0으로 돌아가지 않고, 항상 특정 스레드가 서비스될 수 있다. 이러한 이유로, 양의 한계(92)가 설정된다. 넓은 양의 한계는 다른 이니시에이터(11)의 QoS 계약의 유효화를 어렵게 하는 한편, 작은 양의 한계를 이용하는 QoS 모델은 상호접속부(13)에 의해 유도된 요청 도착 지터를 처리할 수 없을 수도 있다. 더욱이, QoS 방식 및 타깃 동작이 서비스 지터를 유도하는 더 높은 양의 한계가 보장될 수도 있다. 그러므로 양의 한계의 동적 조정이 필요하다.

[0036] 음의 한계(93) 또한 설정된다. 음의 한계(93)는 스레드가 너무 많은 요청을 서비스하고 그만큼 할당된 대역폭을 초과하는 것을 막는다. 이러한 경우라면, 스레드는 음의 할당 카운트(90)의 결과 항상 강등되고 있기 때문에 장시간 동안 서비스를 수신할 수 없다. 그러므로 음의 한계(93)는 서비스[DEW3] 지터를 감소시킨다.

[0037] 도 7은 할당 카운트(90)를 이용하여 특정 스레드에 대한 우선순위를 설정하는 것을 설명하는 흐름도이다. 프로세스(100)는 특정 스레드에 대한 우선순위 부여를 설명한다. 이 우선순위는 타깃(12)이 언제 스레드(21)를 서비스할지를 결정하는데 사용될 수 있다. 블록(101)에서 양의 할당을 갖는 스레드로부터의 요청이 있는지 여부가 판단된다. 할당 카운트(90)에 관련하여 설명한 기술을 이용하여 양의 할당이 결정되고 누적된다. 양의 할당을 갖는 스레드가 발견되면, 블록(102)에서 양의 할당을 갖는 스레드들 중에서 가장 높은 우선순위의 스레드가 선택된다. 일 실시예에 따르면, 가장 높은 우선순위의 스레드는 높은 우선순위 스레드인 스레드 또는 가장 큰 양의 할당 카운트(90)를 갖는 스레드일 수도 있다. 다른 실시예에 따르면, 다른 우선순위 시스템이 설정될 수도 있다.

[0038] 블록(103)에서는, 양의 할당을 갖는 스레드가 없다면, 가장 높은 우선순위의 스레드가 선택되어 서비스된다. 상기와 같이, 우선순위는 높은 우선순위 지정을 갖는 스레드에 서비스를 부여하는 것을 포함하는 다양한 기술을 이용하여 결정될 수 있다. 할당 우선순위는 QoS 유닛(27)에 구현될 수 있다. QoS 유닛(27)이 가장 높은 우선순위를 갖는 스레드를 결정했다면, 그 스레드가 서비스된다. 프로세스는 추후의 서비스를 위해 계속될 수

있다.

[0039] 도 8은 조정 가능한 양의 한계를 이용한 할당 카운트의 동작을 설명하는 흐름도이다. 예를 들어, 높은 우선순위의 스레드가 타깃(12)을 독점하고 있다면, 양의 한계(92)가 조정될 필요가 있을 수도 있다. 타깃(12)에 액세스할 필요가 있는 대역폭 할당 스레드는 높은 우선순위의 스레드 때문에 장시간 동안 그대로 대기할 수도 있다. 결국, 대역폭 할당 스레드의 할당 카운트(90)는 양의 한계(92)에 이르게 된다. 그러나 대역폭 할당 스레드는 여전히 서비스되지 않았다. 한계가 그대로 일정하다면, 어떤 경우에는 QoS 계약이 충족될 수 없기 때문에, 이러한 상황에서 한계(92)를 동적으로 증가하는 것이 어떤 경우에는 바람직할 수도 있다.

[0040] 프로세스(110)는 타깃(12)에 의해 요청이 서비스되어야 할 때마다 실행될 수 있다. 블록(111)에서는, 할당이 양수이면서 높은 우선순위의 스레드로부터의 높은 우선순위 요청이 서비스되는지 여부를 판단한다. 높은 우선순위 스레드로부터 요청이 서비스되었다면, 더 낮은 우선순위의 모든 스레드의 양의 한계(92)가 그 할당 레이트에 비례하는 양만큼 증가한다. 예를 들어, 대역폭 할당 스레드가 50%의 할당 레이트를 갖는다면(즉, 스레드에 특정 타깃(12) 대역폭의 50%가 할당된다면), 한계(92)는 높은 우선순위 스레드에 의해 소비되는 크레디트의 50% 만큼 상승한다. 예를 들어, 일 실시예에서 양의 한계(92)는 6일 수 있다. 높은 우선순위의 스레드가 2개의 크레디트를 소비하면, 한계는 7로 상승하게 된다. 블록(111)으로 돌아가서, 높은 우선순위의 요청이 서비스되지 않았다면, 프로세스(110)는 블록(113)으로 진행한다.

[0041] 블록(113)에서는, 할당 카운트가 양수이면서 높은 우선순위의 요청이 더 많은 할당을 수신했는지 여부를 판단한다. 높은 우선순위 스레드의 할당 카운트가 양의 할당 카운트를 갖는 동시에 증가했다면, 이는 더 높은 우선순위 스레드가 서비스를 요청하지 않고 있다는 표시이다. 더 높은 우선순위의 스레드가 서비스를 요청하고 있지 않기 때문에, 더 낮은 우선순위 스레드가 서비스를 수신하여 그 할당 카운트(90)를 감소시킬 수 있다. 더 낮은 우선순위 스레드가 서비스되고 있거나 서비스를 요청하지 않고 있지만, 어떤 경우에도 양의 한계(92)는 정상으로 돌아가 블록(114)에서 필요 없는 스레드에 우선순위를 부여하는 것을 피해야 한다.

[0042] 본 발명은 특정 실시예들을 참조로 설명하였다. 그러나 발명의 보다 넓은 진의 및 범위를 벗어나지 않으면서 상기 실시예들에 다양한 변형 및 변경이 이루어질 수 있다는 본 개시의 이익은 당업자에게 명백할 것이다. 이에 따라 명세서 및 도면은 한정의 의미보다는 예시로 간주해야 한다.

도면의 간단한 설명

[0007] 도 1은 본 발명의 일 실시예에 따른 예시적인 시스템-온-칩을 나타낸다.

[0008] 도 2는 본 발명의 일 실시예에 따른 QoS 모델을 구현하는 시스템을 나타낸다.

[0009] 도 3은 본 발명의 일 실시예에 따른 QoS 모델을 설명하는 흐름도이다.

[0010] 도 4a~4c는 본 발명의 일 실시예에 따른 도착 모델을 나타낸다.

[0011] 도 5a 및 5b는 본 발명의 일 실시예에 따른 서비스 모델을 나타낸다.

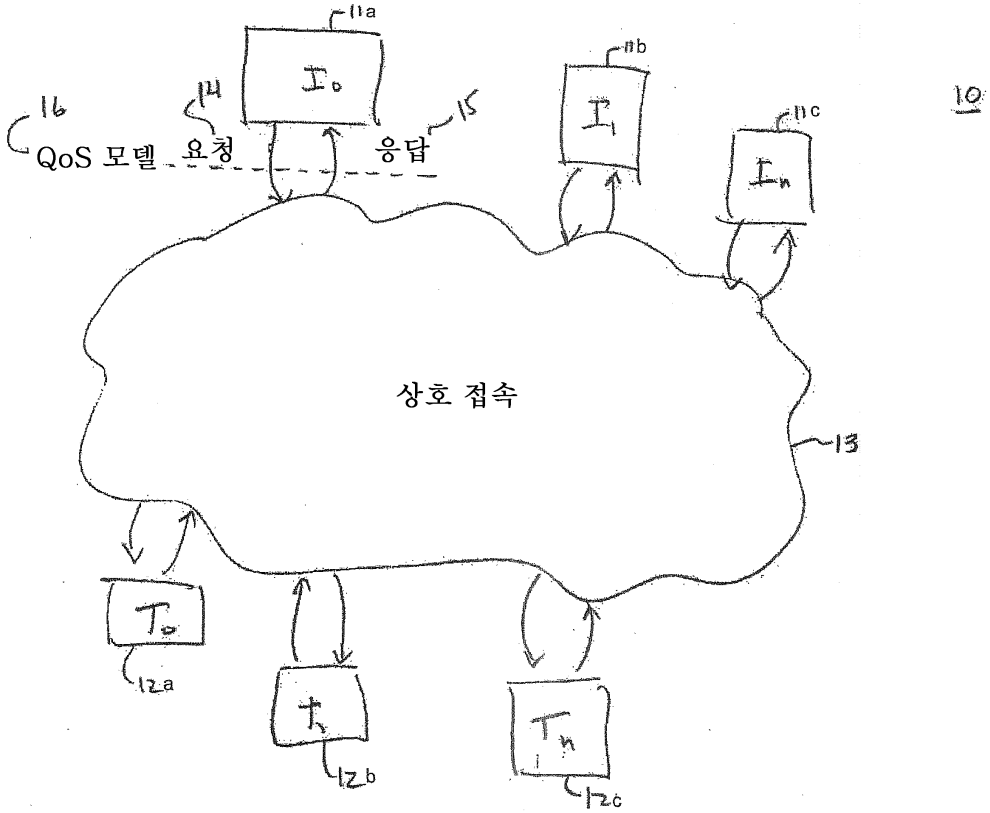
[0012] 도 6은 본 발명의 일 실시예에 따른 할당 카운트를 나타낸다.

[0013] 도 7은 할당 카운트를 이용하여 특정 스레드에 대한 우선순위를 설정하는 것을 설명하는 흐름도이다.

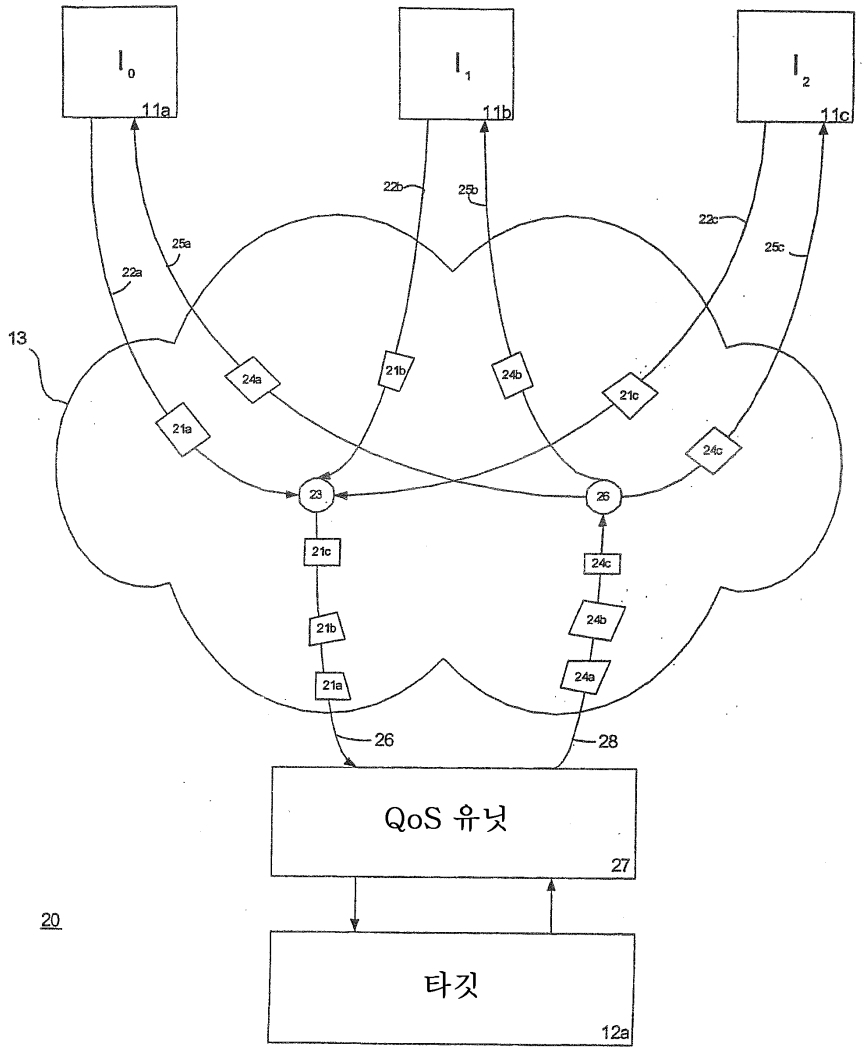
[0014] 도 8은 조정 가능한 양의 한계를 이용한 할당 카운트의 동작을 설명하는 흐름도이다.

도면

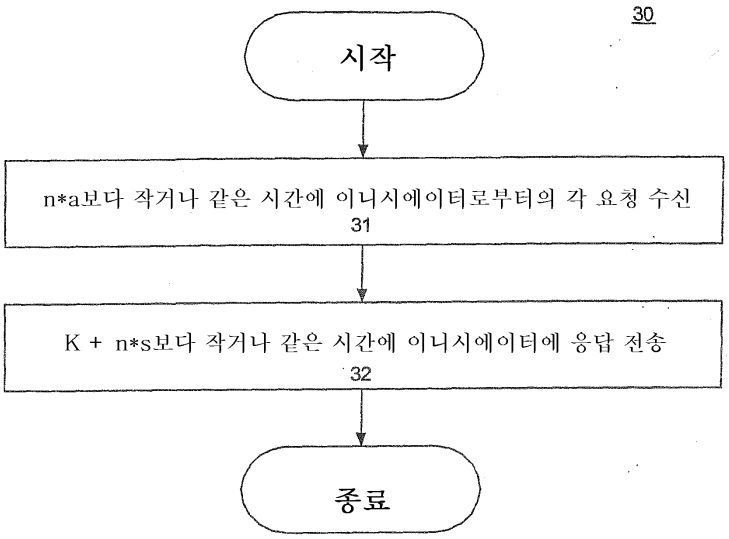
도면1



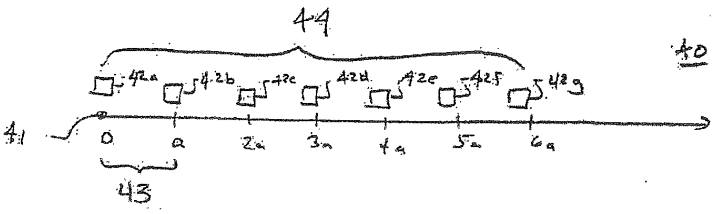
도면2


I0
11a
I1
11b
I2
11c
22b
25b
22c
25c
25a
22a
13
24a
21b
24b
21c
21a
24c
23
26
21c
24c
21b
24b
21a
24a
26
28
QoS 유닛
27
20
타깃
12a

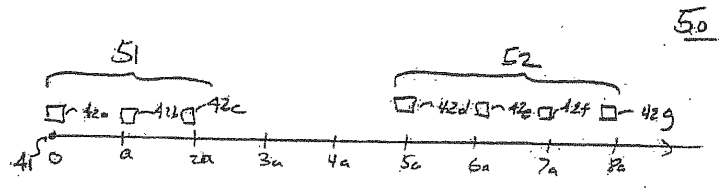
도면3



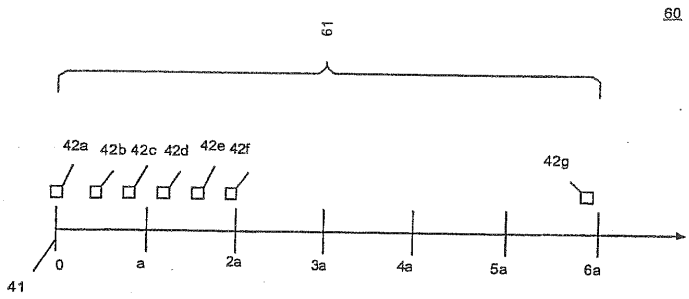
도면4a



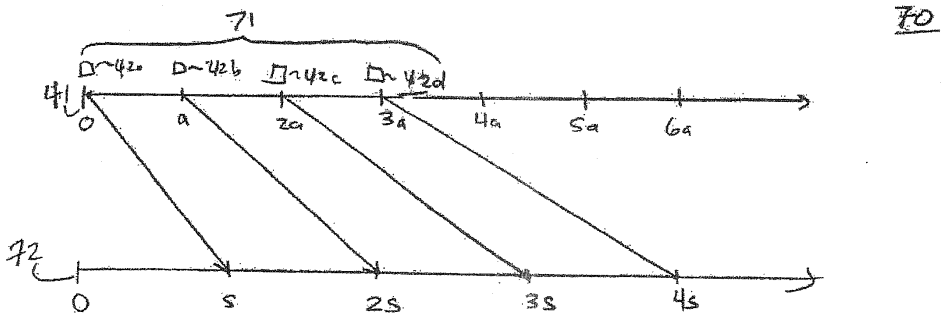
도면4b



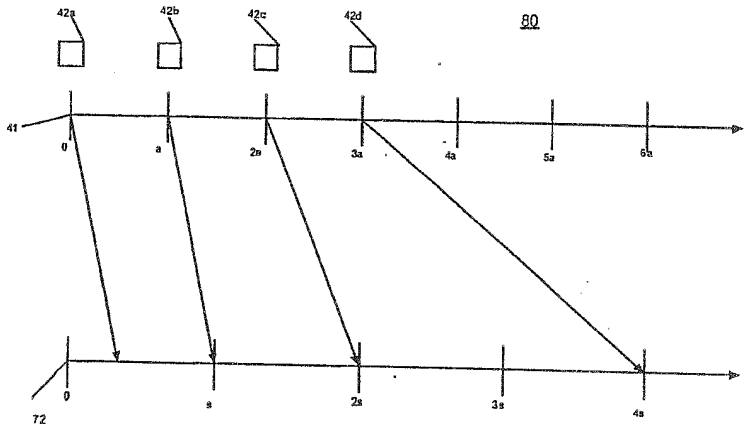
도면4c



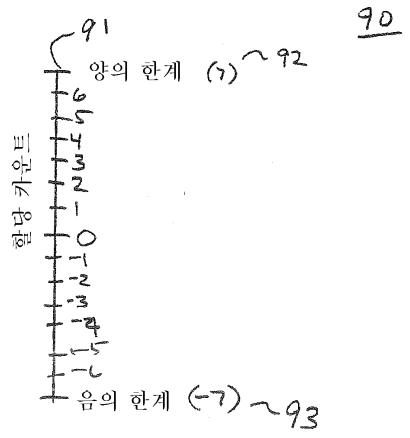
도면5a



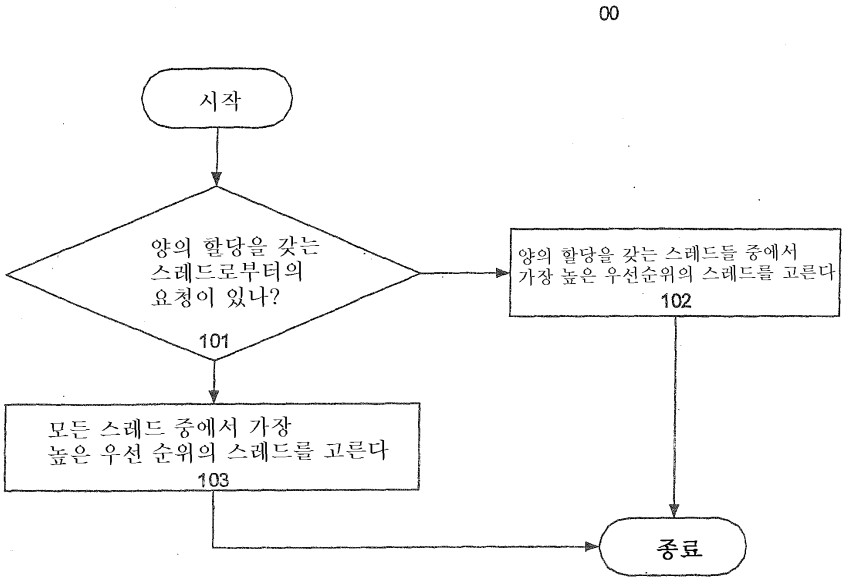
도면5b



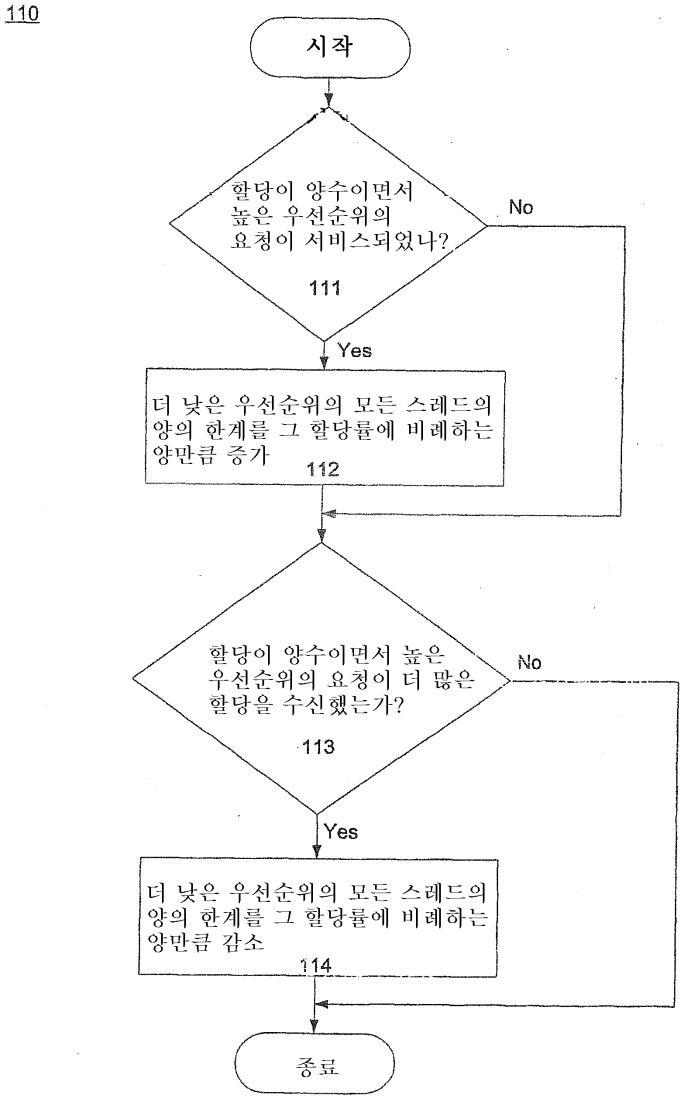
도면6


90
91
양의 한계 (7)
92
할당 카운트
6
5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
음의 한계 (-7)
93

도면7


00
시작
양의 할당을 갖는 스레드로부터의 요청이 있나?
101
양의 할당을 갖는 스레드들 중에서 가장 높은 우선순위의 스레드를 고른다
102
모든 스레드 중에서 가장 높은 우선 순위의 스레드를 고른다
103
종료

도면8


110
시작
할당이 양수이면서 높은 우선순위의 요청이 서비스되었나?
111
No
Yes
더 낮은 우선순위의 모든 스레드의 양의 한계를 그 할당률에 비례하는 양만큼 증가
112
할당이 양수이면서 높은 우선순위의 요청이 더 많은 할당을 수신했는가?
113
No
Yes
더 낮은 우선순위의 모든 스레드의 양의 한계를 그 할당률에 비례하는 양만큼 감소
114
종료