



등록특허 10-2072200



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2020년01월31일
(11) 등록번호 10-2072200
(24) 등록일자 2020년01월23일

- (51) 국제특허분류(Int. Cl.)
G03F 7/20 (2006.01) *G05B 19/418* (2006.01)
G06F 9/48 (2018.01) *H01J 37/317* (2006.01)
B82Y 10/00 (2017.01) *B82Y 40/00* (2017.01)
- (52) CPC특허분류
G03F 7/70525 (2013.01)
G03F 7/70483 (2013.01)
- (21) 출원번호 10-2017-7029925(분할)
- (22) 출원일자(국제) 2012년04월23일
심사청구일자 2017년11월16일
- (85) 번역문제출일자 2017년10월17일
- (65) 공개번호 10-2017-0120194
- (43) 공개일자 2017년10월30일
- (62) 원출원 특허 10-2013-7030924
원출원일자(국제) 2012년04월23일
심사청구일자 2016년12월19일
- (86) 국제출원번호 PCT/EP2012/057366
- (87) 국제공개번호 WO 2012/143548
국제공개일자 2012년10월26일
- (30) 우선권주장
61/478,117 2011년04월22일 미국(US)
61/533,673 2011년09월12일 미국(US)

(56) 선행기술조사문헌

KR1020080103570 A*

US20070203873 A1*

*는 심사관에 의하여 인용된 문헌

전체 청구항 수 : 총 16 항

심사관 : 이석주

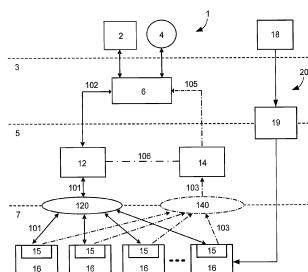
(54) 발명의 명칭 리소그래피 머신들의 클러스터를 위한 네트워크 아키텍처 및 프로토콜

(57) 요약

클러스터링된 기판 프로세싱 시스템은 하나 또는 그 초과의 리소그래피 엘리먼트들을 포함하고, 각각의 리소그래피 엘리먼트는 패턴 데이터에 따른 기판들의 독립적인 노출을 위해 배열된다. 각각의 리소그래피 엘리먼트는 복수의 리소그래피 서브시스템들을 포함하고, 제어 네트워크는 상기 리소그래피 서브시스템들과 적어도 하나의 엘

(뒷면에 계속)

대 표 도



리먼트 제어 유닛 사이의 제어 정보의 통신을 위해 배열되고, 상기 엘리먼트 제어 유닛은 명령들을 상기 리소그래피 서브시스템들에 전송하도록 배열되고, 그리고 상기 리소그래피 서브시스템들은 응답들을 상기 엘리먼트 제어 유닛에 전송하도록 배열된다. 각각의 리소그래피 엘리먼트는 또한, 오퍼레이터 또는 호스트 시스템에 인터페이스하기 위해 클러스터 프론트-엔드를 포함하고, 상기 프론트-엔드는 하나 또는 그 초과의 웨이퍼들의 노출을 위해 상기 하나 또는 그 초과의 리소그래피 서브시스템들의 동작을 제어하기 위해 상기 적어도 하나의 엘리먼트 제어 유닛에 제어 정보를 발행하기 위해 배열된다. 상기 프론트-엔드는 상기 엘리먼트 제어 유닛에 프로세스 프로그램을 발행하기 위해 배열되고, 상기 프로세스 프로그램은 미리규정된 명령들 및 연관된 파라미터들의 세트를 포함하고, 각각의 명령은, 상기 리소그래피 서브시스템들 중 하나 또는 그 초과에 의해 수행될 미리규정된 동작 또는 동작들의 시퀀스에 대응하고, 그리고 상기 파라미터들은 상기 동작 또는 동작들의 시퀀스가 수행될 방법을 추가로 규정한다.

(52) CPC특허분류

G03F 7/70508 (2013.01)

G03F 7/70991 (2013.01)

G05B 19/41865 (2013.01)

G06F 9/48 (2013.01)

H01J 37/3177 (2013.01)

B82Y 10/00 (2013.01)

B82Y 40/00 (2013.01)

명세서

청구범위

청구항 1

하나 이상의 리소그래피 엘리먼트들을 포함하는 리소그래피 시스템에 있어서,

각각의 리소그래피 엘리먼트는 패턴 데이터에 따라 기관들을 독립적으로 노출하도록 구성되고, 각각의 리소그래피 엘리먼트는 복수의 리소그래피 서브시스템들을 포함하고,

상기 리소그래피 시스템은, 제어 정보의 통신을 위해 상기 복수의 리소그래피 서브시스템들과 적어도 하나의 엘리먼트 제어 유닛 사이에 제어 네트워크 경로를 형성하는 제어 네트워크를 더 포함하고,

상기 리소그래피 시스템은:

하나 이상의 웨이퍼들의 노출을 위해 상기 리소그래피 서브시스템들 중 하나 이상의 동작을 제어하기 위해, 제어 정보를 상기 적어도 하나의 엘리먼트 제어 유닛에 발행(issue)하고;

상기 엘리먼트 제어 유닛에 프로세스 프로그램을 발행하도록

구성되며,

상기 프로세스 프로그램은 미리규정된 명령들 및 연관된 파라미터들의 세트를 포함하고, 각각의 명령은 상기 리소그래피 서브시스템들 중 하나 이상에 의해 수행될 미리규정된 동작 또는 동작들의 시퀀스에 대응하고, 그리고 상기 파라미터들은 상기 동작 또는 동작들의 시퀀스가 수행될 방법을 더 규정하고,

상기 프로세스 프로그램 내의 미리규정된 명령들의 세트는 상기 리소그래피 서브시스템들 중 하나 이상의 업데이팅 또는 업그레이딩 소프트웨어를 포함하고, 상기 서브시스템들 각각은 상기 서브시스템에 로컬라이징된 기능성을 갖는 소프트웨어로 업데이팅 또는 업그레이딩 가능한 것을 특징으로 하는,

리소그래피 시스템.

청구항 2

제1항에 있어서,

상기 엘리먼트 제어 유닛은, 상기 리소그래피 서브시스템들 중 하나 이상에 의해 실행되기 위한 프로세스 잡(process job)을 생성하도록 상기 프로세스 프로그램을 스케줄링하도록 구성되는,

리소그래피 시스템.

청구항 3

제2항에 있어서,

상기 소프트웨어를 업데이트 또는 업그레이드하기 위한 리소그래피 서브시스템 내의 소프트웨어의 수정(modification)은, 프로세스 잡을 실행함으로써 수행되는,

리소그래피 시스템.

청구항 4

제1항 내지 제3항 중 어느 한 항에 있어서,

상기 복수의 리소그래피 서브시스템들은, 중앙 디스크 또는 메모리 내의 상기 리소그래피 서브시스템을 위한 부트 이미지(boot image)를 업데이트함으로써 리소그래피 서브시스템 내의 소프트웨어의 수정이 수행되도록, 상기 중앙 디스크 또는 메모리로부터 부팅(boot)하도록 구성되는,

리소그래피 시스템.

청구항 5

제1항 내지 제3항 중 어느 한 항에 있어서,

상기 리소그래피 시스템은, 데이터를 수집하고 저장하고 관리하기 위한 데이터 네트워크, 및 상기 데이터 네트워크를 통해 상기 복수의 리소그래피 서브시스템들에 연결된 데이터 네트워크 허브를 더 포함하고,

상기 복수의 리소그래피 서브시스템들은, 상기 데이터 네트워크 허브 상에서 상기 리소그래피 서브시스템을 위한 부트 이미지를 업데이트함으로써 상기 소프트웨어의 업데이트 또는 업그레이드가 수행되도록, 상기 데이터 네트워크 허브로부터 부팅하도록 구성되는,

리소그래피 시스템.

청구항 6

제1항 내지 제3항 중 어느 한 항에 있어서,

상기 엘리먼트 제어 유닛은, 상기 프로세스 프로그램을 스케줄링하고, 그리고 상기 리소그래피 서브시스템으로 전송된 선행(preceding) 명령의 실행 상태와 상관없이, 상기 스케줄링에 따라 리소그래피 서브시스템에 상기 프로세스 프로그램의 명령을 전송하도록 구성되는,

리소그래피 시스템.

청구항 7

제1항 내지 제3항 중 어느 한 항에 있어서,

상기 리소그래피 시스템을 새로운 기능성으로 확장 또는 업그레이드하는 것은, 상기 리소그래피 서브시스템들 중 하나 이상에서의 소프트웨어의 수정에 의해, 그리고 업데이트된 프로세스 프로그램을 발행함으로써 수행되는,

리소그래피 시스템.

청구항 8

제7항에 있어서,

상기 리소그래피 서브시스템들 중 하나 이상은, 상기 리소그래피 서브시스템들 중 하나 이상에서의 소프트웨어의 수정 이후에 상기 엘리먼트 제어 유닛에 실행가능한 명령들의 정보를 전송하도록 구성되는,

리소그래피 시스템.

청구항 9

제7항에 있어서,

상기 리소그래피 시스템의 확장 또는 업그레이드는 상기 엘리먼트 제어 유닛 내의 소프트웨어의 수정을 필요로 하지 않는,

리소그래피 시스템.

청구항 10

제1항 내지 제3항 중 어느 한 항에 있어서,

상기 복수의 리소그래피 서브시스템들의 설정(setting)들을 포함하는 데이터를 수집하고 저장하고 관리하기 위한 데이터 네트워크를 더 포함하고,

상기 리소그래피 서브시스템들은 상기 데이터 네트워크를 통해 데이터 네트워크 허브에 연결되는,

리소그래피 시스템.

청구항 11

제10항에 있어서,

상기 제어 네트워크 및 데이터 네트워크는 별개의 네트워크들을 형성하는,
리소그래피 시스템.

청구항 12

제10항에 있어서,
상기 데이터 네트워크 허브에 의해 저장되는 데이터에는 타임스탬프(timestamp)가 태그(tag)되는,
리소그래피 시스템.

청구항 13

제10항에 있어서,
상기 패턴 데이터를 생성하기 위한 패턴 데이터 프로세싱 시스템을 더 포함하고,
상기 패턴 데이터 프로세싱 시스템은, 상기 제어 네트워크 및 데이터 네트워크와 구별되는 데이터 경로를 통해,
상기 패턴 데이터를 상기 리소그래피 서브시스템들 중 하나 이상으로 전송하도록 구성되는,
리소그래피 시스템.

청구항 14

제1항 내지 제3항 중 어느 한 항에 있어서,
상기 엘리먼트 제어 유닛은 반복되는 명령들의 시퀀스를 상기 리소그래피 서브시스템에 전송하도록 구성되고,
상기 리소그래피 서브시스템은 상기 시퀀스 내의 명령이 시간 기간 내에 실행되지 않은 경우 상기 시퀀스 내의
명령을 폐기(discard)하도록 구성되는,
리소그래피 시스템.

청구항 15

제1항 내지 제3항 중 어느 한 항에 있어서,
상기 프로세스 프로그램은 제1 명령에 대한 제1의 미리결정된 시간 기간을 규정하고,
상기 엘리먼트 제어 유닛은, 상기 시간 기간이 만료될 때까지 그리고 상기 제1 명령의 실행 상태와 무관하게,
상기 제1 명령을 뒤따르는 다음 명령을 개시(initiate)하는 것을 지연시키도록 구성되는,
리소그래피 시스템.

청구항 16

제1항 내지 제3항 중 어느 한 항에 있어서,
상기 제어 네트워크는 실시간 통신 프로토콜을 사용하지 않고 준(quasi)-실시간 수행을 제공하는,
리소그래피 시스템.

발명의 설명

기술 분야

[0001] 본 발명은 패턴 데이터에 따른 기판들의 독립적인 노출을 위해 각각 배열된 복수의 리소그래피 엘리먼트들을 포함하는 클러스터링된 기판 프로세싱 시스템, 그리고 보다 구체적으로는 이러한 클러스터링된 기판 프로세싱 시스템을 위한 네트워크 아키텍처 및 프로토콜에 관한 것이다.

배경 기술

[0002] 반도체 산업에서, 높은 정확도 및 신뢰도로 구조들을 제조하기 위한 계속 증가하는 욕구가 존재한다. 리소그래

피 시스템들에서, 이러한 욕구는 웨이퍼들의 생산의 양 및 속도에 관하여 극히 높은 주문들을 초래한다. 리소그래피 머신들은, 머신 내에서 다양한 기능들을 수행하기 위한 많은 서브시스템들, 그리고 서브시스템들의 동작을 제어하기 위한 복잡한 소프트웨어로 더욱 복잡해진다. 현대의 리소그래피 머신들은 통상적으로 많은 양들의 데이터 동작을 발생시키고, 그리고 데이터를 관리하는 것은 점점 더 어려워진다. 리소그래피 머신들은 더 높은 볼륨 생산을 제공하기 위해 리소그래피 엘리먼트들의 클러스터로 함께 그룹화될 수 있다. 제어 및 데이터 수집을 위한 이러한 리소그래피 엘리먼트들의 클러스터 내에서의 정보의 송신은, 전송할 정보의 양이 증가하고 시스템들의 복잡도가 증가함에 따라 문제가 된다.

[0003] 제어 및 데이터 서비스들을 단일 네트워크 내로 결합시키고, 중요 제어 데이터의 적시의(timely) 전달을 제공하도록 특정 데이터의 전달을 우선순위화하기 위해 서비스 품질 기법들을 이용하는 네트워크 아키텍처들이 존재하지만, 이러한 환경에서 불충분한 것으로 증명되었다. 본 발명은 리소그래피 엘리먼트들의 클러스터를 위한 많은 양들의 데이터의 수집 및 중요 제어 데이터의 적시의 전달을 제공하기 위해 네트워크 아키텍처에 대한 대안적인 솔루션을 제공한다.

발명의 내용

[0004] 복수의 리소그래피 머신들을 클러스터링하는 것이 바람직하며, 제어 및 데이터 서비스들은 클러스터 네트워크 아키텍처 및 프로토콜 내에 수용된다. 일 양상에서 본 발명은, 하나 또는 그 초과의 리소그래피 엘리먼트들을 포함하는 클러스터링된 기관 프로세싱 시스템을 제공하고, 각각의 리소그래피 엘리먼트는 패턴 데이터에 따른 기관들의 독립적인 노출을 위해 배열된다. 각각의 리소그래피 엘리먼트는, 복수의 리소그래피 서브시스템들, 리소그래피 서브시스템들과 적어도 하나의 엘리먼트 제어 유닛 사이의 제어 정보의 통신을 위해 배열된 제어 네트워크 – 상기 엘리먼트 제어 유닛은 리소그래피 서브시스템들에 명령들을 전송하도록 배열되고, 리소그래피 서브시스템들은 엘리먼트 제어 유닛에 응답들을 전송하도록 배열됨 – 를 포함한다. 각각의 리소그래피 엘리먼트는 또한, 오퍼레이터 또는 호스트 시스템에 인터페이스하기 위한 클러스터 프론트-엔드(front-end) – 상기 프론트-엔드는, 하나 또는 그 초과의 웨이퍼들의 노출을 위해 하나 또는 그 초과의 리소그래피 서브시스템들의 동작을 제어하기 위해 적어도 하나의 엘리먼트 제어 유닛에 제어 정보를 발행(issue)하기 위해 배열됨 – 를 포함한다. 프론트-엔드는 엘리먼트 제어 유닛에 프로세스 프로그램을 발행하기 위해 배열되고, 프로세스 프로그램은 미리규정된 명령들 및 연관된 파라미터들의 세트를 포함하고, 각각의 명령은 리소그래피 서브시스템들 중 하나 또는 그 초과에 의해 수행될 미리규정된 동작 또는 동작들의 시퀀스에 대응하고, 그리고 파라미터들은 동작 또는 동작들의 시퀀스가 수행될 방법을 더 규정한다.

[0005] 엘리먼트 제어 유닛은, 리소그래피 서브시스템들에 의한 실행을 위해 대응하는 프로세스 잡(job)을 발생시키기 위해 프로세스 프로그램을 스케줄링하기 위해 배열될 수 있고, 프로세스 잡은 프로세스 프로그램의 명령들의 세트 및 명령들 각각을 위해 스케줄링된 실행 시간을 포함한다. 프로세스 잡은 프로세스 잡의 실행이 시작하기 전에 완전히 스케줄링될 수 있다. 이는, 완전한 프로세스 잡이 시작하기 전에, 완전한 프로세스 잡의 실행을 위한 알려진 그리고 고정된 시간 기간을 초래한다. 각각의 단계를 위한 그리고 완전한 프로세스의 스케줄링된 시작 시간(및 완료 시간)은 클러스터 프론트-엔드에 보고될 수 있다. 이는, 기관들을 리소그래피 엘리먼트에 그리고 리소그래피 엘리먼트로부터 전달하는 트랙 시스템을 스케줄링하는 것을 용이하게 하는, 기관들이 리소그래피 엘리먼트에 전달되어야 할 때 그리고 리소그래피 엘리먼트 내에서의 기관의 노출이 완료될 것으로 예상될 때와 같은, 패(fab) 내의 활동들의 스케줄링을 매우 단순화한다.

[0006] 엘리먼트 제어 유닛은 프로세스 프로그램의 명령들의 세트를 포함하는 프로세스 잡을 발생시키기 위해 배열될 수 있고, 그리고 명령들 각각을 위해, 리소그래피 서브시스템의 아이덴티티는 명령을 실행시키기 위해 스케줄링된다. 엘리먼트 제어 유닛은, 프로세스 잡의 명령들 각각을, 명령들 각각을 위해 스케줄링된 실행 시간에, 식별된 리소그래피 서브시스템에 전송함으로써 프로세스 잡을 실행시키도록 배열될 수 있다. 엘리먼트 제어 유닛은, 프로세스 잡의 선행(preceding) 명령의 실행 상태와 상관없이 명령들 각각을 위해 스케줄링된 실행 시간에, 프로세스 잡의 명령들 각각을, 식별된 리소그래피 서브시스템에 전송하도록 배열될 수 있다. 프로세스 잡의 명령들의 실행 시간은, 선행 또는 동시의 단계의 결과들에 조건적이지 않지만, 미리결정된 시간 스케줄에 따라 실행된다. 단계가 정확하게 또는 스케줄링된 시간 내에 완료되지 않는 경우, 이는 다시 실행되지 않을 것이지만, 대신에 다음 스케줄링된 단계가 실행될 것이다.

[0007] 프로세스 프로그램은 대응하는 명령을 위한 미리결정된 시간 기간을 규정할 수 있고, 그리고 엘리먼트 제어 유닛은, 리소그래피 서브시스템들에 의한 실행을 위해 프로세스 잡을 발생시키도록 프로세스 프로그램을 스케줄링하기 위해 배열될 수 있고, 프로세스 잡은 각각의 명령을 위해 스케줄링된 실행 시간을 포함하고, 미리결정된

시간 기간은 프로세스 잡 내 대응하는 명령을 위해 스케줄링된 시간을 결정하기 위해 이용된다.

[0008] 프로세스 프로그램은 대응하는 제 1 명령을 위한 제 1 미리결정된 시간 기간을 규정할 수 있고, 엘리먼트 제어 유닛은, 시간 기간의 만료까지 그리고 제 1 명령의 실행 상태와 상관없이, 제 1 명령을 뒤따르는 다음 명령을 개시하는 것을 지연시키도록 배열될 수 있다. 프로세스 잡의 명령들의 실행의 스케줄링된 타이밍은 성공적인 완료 또는 심지어 선행 단계로부터의 명령을 실행하는 것의 실패에 관한 서브시스템들로부터의 피드백에 따르지 않는다. 서브시스템이, 명령의 실행이 완료되었다는 것을 보고하는 경우, 엘리먼트 제어 유닛은 다음 명령을 실행하기 전에 다음 명령을 위해 스케줄링된 실행 시간까지 대기하도록 배열된다. 서브시스템이, 명령을 실행하는 데 있어서의 실패 또는 에러를 보고하는 경우, 엘리먼트 제어 유닛은, 다음 명령을 위해 스케줄링된 실행 시간까지 대기하며, 그 다음으로, 다음 명령을 실행하는 것을 계속할 것이다. 프로세스 프로그램의 명령의 실행에 있어서의 임의의 타이밍 변화가, 프로세스 잡의 스케줄링에서 고려되어서, 명령의 실행을 위해 스케줄링된 시간은 가장 크게 예상된 실행 시간보다 더 크다. 시간 스케줄은 통상적으로 프로세스 잡에서 규정된다.

[0009] 엘리먼트 제어 유닛은, 동작 또는 동작들을 위한 하나 또는 그 초과의 파라미터들을 제어 네트워크를 통해 서브시스템에 전송함으로써, 그리고 후속하여 동작 또는 동작들에 대응하는 명령을 서브시스템에 전송함으로써, 리소그래피 서브시스템들 중 하나에 의해 동작 또는 동작들의 시퀀스를 개시하도록 배열될 수 있다. 엘리먼트 제어 유닛은, 시간 기간에 의해 명령의 실행을 위해 스케줄링된 시간보다 미리 하나 또는 그 초과의 파라미터들을 서브시스템에 전송하도록 배열될 수 있고, 시간 기간은, 명령이 서브시스템에 전송되기 전에, 서브시스템이 하나 또는 그 초과의 파라미터들을 수신하는 것을 보장하기에 충분하다. 하나 또는 그 초과의 파라미터들이 하나 또는 그 초과의 메시지들에서 전송될 수 있어서, 임의의 하나의 메시지는 크기가 제한된다. 파라미터들은 상당한 양의 데이터를 포함할 수 있어서, 하나 또는 그 초과의 파라미터들을 포함하는 더 큰 메시지들은, 서브시스템들로의 송신의 타이밍이 중요하지 않을 때 미리 전송된다. (연관된 파라미터들 없이) 단지 명령만을 포함하는 훨씬 더 작은 메시지는 명령의 실행을 위해 스케줄링된 시간에서 전송될 수 있다. 이러한 대책은, 송신 시간을 감소시키고, 그리고 명령의 적시의 그리고 신뢰적인 송신을 증가시키기 위해, 명령이 전송될 때 혼잡을 회피하고, 네트워크 상의 부하를 분산시킨다.

[0010] 엘리먼트 제어 유닛에 의해 발행된 명령에 대응하는 동작 또는 동작들의 시퀀스의 완료시에, 동작 또는 동작들의 시퀀스를 완료한 리소그래피 서브시스템은, 완료를 엘리먼트 제어 유닛에 통지하기 위해, 그리고 엘리먼트 제어 유닛으로부터의 인스트럭션을 수신시에, 동작 또는 동작들의 시퀀스의 실행에 관한 데이터를 발행하도록 배열될 수 있다. 엘리먼트 제어 유닛이, 동작 또는 동작들의 시퀀스의 완료의 통지를 서브시스템으로부터 수신 할 때, 엘리먼트 제어 유닛은 동작 또는 동작들의 시퀀스의 실행에 관한 데이터를 전송하라는 요청을 서브시스템에 전송할 수 있다. 이는, 송신 시간을 감소시키고, 그리고 적시의 그리고 신뢰적인 송신을 증가시키기 위해, 완료 통지가 전송될 때 혼잡을 회피하고, 네트워크 상의 부하를 분산시키고, 그리고 그 다음으로 데이터는, 후에 타이밍이 중요하지 않을 때 검색될 수 있다.

[0011] 프로세스 프로그램은 어떠한 조건적인 단계들도 포함하지 않도록 프로그래밍될 수 있다. 프로세스 프로그램은 대안적인 명령들로서 프로그래밍된 조건적인 단계들을 포함할 수 있고, 대안적인 명령들은 프로세스 잡과 동시에 스케줄링되고, 상기 대안적인 명령들 각각은 동일한 실행시간을 할당받아서, 전체로서 프로세스 잡의 실행 시간은 어느 대안적인 명령이 실행을 위해 선택되는지에 따라 변화하지 않는다.

[0012] 리소그래피 서브시스템들은 엘리먼트 제어 유닛으로부터의 명령의 수신을 확인응답하기 위해 엘리먼트 제어 유닛에 확인응답을 전송하도록 배열될 수 있다. 확인응답은, 서브시스템이 예상되는 바와 같이 응답하는 것을 중단할 때를 검출하기 위해 엘리먼트 제어 유닛에 의해 이용될 수 있다. 확인응답은, 서브시스템이 관련 명령을 수신하였고, 여전히 기능하고 있다는 확인(confirmation), 확인응답을 전송하는 서브시스템의 아이덴티티, 및 시간 스템프 이외의 어떠한 다른 정보도 반송(carry)하지 않도록 구성될 수 있다. 서브시스템들은 프로세스 잡에 대한 명령을 실행하기 위한 어떠한 타임아웃들도 없이 배열될 수 있다. 엘리먼트 제어 서브시스템에서의 타임아웃은, 서브시스템들로부터의 확인응답에 대한 예상되는 응답 시간보다 매우 더 긴 시간 기간으로 설정될 수 있다. 이러한 타임아웃은, 서브시스템이 기능하지 않는다는 합리적인 의심의 여지없다는(beyond reasonable doubt) 결론을 내리기에는 충분히 크지만, 오퍼레이터가 정상적으로 그렇게 하기 전에 실패를 검출하기에 충분히 작다.

[0013] 소프트웨어를 업데이트 또는 업그레이드하기 위한 서브시스템 내 소프트웨어의 수정은 엘리먼트 제어 유닛 내에서 프로세스 잡을 실행시킴으로써 수행된다. 이러한 수정은 서브시스템의 유틸리티 소프트웨어 또는 임의의 애플리케이션 또는 서브시스템의 기본 운영 체제(operating system) 소프트웨어를 수정하는 것을 포함할 수 있다.

이는, 서브시스템 소프트웨어의 업데이트 또는 업그레이드와 같은 리소그래피 동작들 및 서비스 동작들 양측 모두를 위해 하나의 단일 인터페이스 방법을 가능하게 한다.

[0014] 제어 네트워크는 실시간 통신 프로토콜을 사용함이 없이, 준(quasi) 실시간 수행을 제공하도록 배열될 수 있다.

[0015] 각각의 리소그래피 엘리먼트는, 리소그래피 서브시스템들로부터 적어도 하나의 데이터 네트워크 허브로의 데이터 로깅 정보의 통신을 위해 배열된 데이터 네트워크를 더 포함할 수 있고, 리소그래피 서브시스템들은 데이터 로깅 정보를 데이터 네트워크 허브에 전송하도록 배열되고, 그리고 데이터 허브는 데이터 로깅 정보를 수신 및 저장하기 위해 배열되고, 그리고 프론트-엔드는 데이터 네트워크 허브에 의해 수신된 데이터 로깅 정보의 적어도 일부를 수신하기 위해 더 배열된다. 이러한 네트워크 아키텍처는 제어 및 데이터 기능들을 별개의 네트워크들로 분리시킨다. 제어 기능들은 양방향 데이터 흐름을 요구하고, 그리고 제어 네트워크를 통한 예측가능한 정보 전달을 요구한다. 데이터 수집 및 관리 기능들은 일반적으로, 리소그래피 서브시스템들로부터 데이터 네트워크 내에서 상방향으로(upward) 데이터 네트워크 허브로, 그리고 전방향으로(onward) 클러스터 인터페이스로, 단방향 데이터 흐름을 요구하고, 그리고 매우 큰 양들의 데이터를 수반할 수 있다. 리소그래피 서브시스템들은 제 1 송신 레이트로 데이터를 데이터 네트워크에 전송하도록 배열될 수 있고, 그리고 데이터 네트워크 허브는 제 2 송신 레이트로 데이터 네트워크로부터 데이터를 수신하도록 배열되고, 제 2 송신 레이트는 제 1 송신 레이트보다 매우 더 크다.

[0016] 제어 네트워크는 TCP 링크 상에서 바이트(byte) 송신 프로토콜을 이용할 수 있다. 엘리먼트 제어 유닛 및 리소그래피 서브시스템들은 제어 네트워크 상에서 메시지 시퀀스들을 전송하도록 배열될 수 있고, 그리고 메시지 시퀀스의 제 1 메시지는 2개의 엘리먼트들을 포함하고, 제 1 엘리먼트는 메시지 타입을 갖는 스트링을 포함하고, 그리고 제 2 엘리먼트는 메시지에 대한 네임드 아규먼트(named argument)들을 갖는 사전(dictionary)을 포함한다. 메시지 시퀀스의 제 1 메시지는 자바스크립트 객체 표기법(JavaScript Object Notation)(JSON)으로 인코딩된 제어 데이터 구조를 포함할 수 있다. 메시지 시퀀스의 제 2 메시지는 하나 또는 그 초과의 인코딩된 데이터 메시지들을 포함할 수 있고, 데이터 메시지들의 인코딩은 제어 네트워크 프로토콜에 의해 설정 또는 제한되지 않는다. 임의의 메시지들을 제어 메시지들과 맵팅하는 이러한 방식은, 프로세스 잡 명령들 및 출력들이, 오퍼레이터에 의해 이해 가능한 데이터 포맷으로 직접적으로 생성되는 것을 허용하고, 그리고 부가적인 인코딩 없이 이러한 메시지들의 송신을 허용한다.

[0017] 제어 네트워크는 또한, 다수의 데이터 전달들이 제어 네트워크 상에서 동시에 발생하지 않는다는 것을 보장하도록 배열된 제어기를 포함할 수 있다.

[0018] 본 발명의 실시예들은 이제, 첨부된 개략도들을 참조하여, 단지 예시로서 기술될 것이다.

도면의 간단한 설명

[0019] 도 1은 본 발명에 따른 리소그래피 시스템을 위한 네트워크 아키텍처의 실시예의 개략도이고,

도 2는 리소그래피 엘리먼트들의 클러스터를 포함하는 리소그래피 시스템을 위한 네트워크 아키텍처의 실시예의 개략도이고,

도 3은 리소그래피 엘리먼트들의 클러스터의 레이아웃의 도면이고,

도 4는 하전 입자(charged particle) 리소그래피 엘리먼트의 전자-광학 컬럼(column)의 간략화된 도면이고,

도 5는 본 발명의 실시예에 따른 리소그래피 툴을 위한 OSI 계층화된 모델에서의 엘리먼트 제어 유닛 프로토콜의 개략도이고,

도 6a는 본 발명의 실시예에 따른 접속 시퀀스의 개략도이고,

도 6b는 본 발명의 실시예에 따른 재접속 시퀀스의 개략도이고,

도 6c는 본 발명의 실시예에 따른 실행 명령 시퀀스의 개략도이고,

도 6d는 본 발명의 실시예에 따른 중단(abort) 시퀀스의 개략도이고,

도 6e는 본 발명의 실시예에 따른 경합 조건(race condition)을 갖는 중단 시퀀스의 개략도이고,

도 6f는 본 발명의 실시예에 따른 포괄적인(generic) 명령 동안의 예외 시퀀스의 개략도이고,

도 6g는 본 발명의 실시예에 따른 자발적(spontaneous) 예외의 개략도이고,

도 6h는 본 발명의 실시예에 따른 실행 명령 동안의 예외의 개략도이다.

발명을 실시하기 위한 구체적인 내용

[0020]

다음은, 단지 예시로서 주어지고, 도면들을 참조하는 본 발명의 특정 실시예들을 기술한다.

[0021]

도 1은 본 발명에 따른 제어 및 데이터 인터페이스들을 갖는 리소그래피 시스템(1)의 일 실시예의 개략도이다. 상기 도 1은, 3개의 인터페이스들, 즉 클러스터 인터페이스(3), 클러스터 엘리먼트 인터페이스(5), 및 리소그래피 서브시스템 인터페이스들(7)을 갖는 계층적 어레인저먼트를 도시한다. 도 1은 다수의 리소그래피 서브시스템들(16)을 포함하는 하나의 리소그래피 엘리먼트(10)를 포함하는 리소그래피 시스템 클러스터를 갖는 구성을 예시한다. 리소그래피 시스템은, 예를 들어 도 2 실시예에서와 같이 다수의 리소그래피 엘리먼트들(10)을 포함할 수 있다.

[0022]

클러스터 인터페이스(3)는 리소그래피 클러스터 프론트-엔드(6)와 하나 또는 그 초과의 호스트 시스템들(2) 사이의, 및/또는 클러스터 프론트-엔드(6)와 하나 또는 그 초과의 오퍼레이터 콘솔들(4) 사이의 통신을 위한 인터페이스들을 포함한다.

[0023]

클러스터 엘리먼트 인터페이스(5)는 엘리먼트 제어 유닛(12) 및/또는 데이터 네트워크 허브(14)를 포함하는 리소그래피 엘리먼트 네트워크와 클러스터 프론트-엔드(6) 사이의 통신을 위한 인터페이스들을 포함한다. 엘리먼트 제어 유닛(12)은 링크(106)를 통해 데이터 네트워크 허브(14)와 통신될 수 있고, 통신은 바람직하게, 엘리먼트 제어 유닛(12)으로부터 데이터 네트워크 허브(14)로의 단방향성이다.

[0024]

리소그래피 서브시스템 인터페이스(7)는 엘리먼트 제어 유닛(12)과 리소그래피 서브시스템들(16) 사이의, 그리고 데이터 네트워크 허브(14)와 리소그래피 서브시스템들(16) 사이의 인터페이스들을 포함한다. 서브시스템들(16)은 제어 네트워크(120)를 통해 엘리먼트 제어 유닛(12)과 통신하고, 그리고 서브시스템들(16)은 데이터 네트워크(140)를 통해 데이터 네트워크 허브(14)와 통신한다.

[0025]

도 2는 리소그래피 시스템 클러스터가 다수의 리소그래피 엘리먼트들(10)을 포함하고, 각각의 엘리먼트는 다수의 리소그래피 서브시스템들(16)을 포함하는 리소그래피 시스템(1)의 실시예의 개략도이다. 각각의 엘리먼트는, 엘리먼트를 위해 리소그래피 서브시스템들(16)과 통신하는 데이터 네트워크 허브(14) 및 엘리먼트 제어 유닛(12)을 포함할 수 있다. 각각의 엘리먼트는, 리소그래피 프로세스를 이용하여 웨이퍼들을 독립적으로 노출시키도록 동작할 수 있는 독립형(stand-alone) 리소그래피 엘리먼트로서 기능할 수 있다. 이러한 실시예에서, 다수의 리소그래피 엘리먼트들(10)은 각각 단일 프론트-엔드(6)와 통신하고, 프론트-엔드(6)는 전체 클러스터를 위해 기능하는 하나 또는 그 초과의 호스트 시스템들(2) 및/또는 오퍼레이터 인터페이스들(4)과 통신한다.

[0026]

도 1 및 도 2의 실시예들은 바람직하게, 리소그래피 엘리먼트들의 클러스터의 효율적인 제어를 용이하게 하도록 설계된다. 각각의 리소그래피 엘리먼트(10)는 바람직하게 단지, 제어 네트워크(120) 및 데이터 네트워크(140)로의 네트워크 인터페이스들만을 갖는다. 이러한 설계 규칙에 대한 예외는, 패턴 스트리머(pattern streamer)(19)를, 하전 입자 빔들을 변조 또는 스위칭하는 것을 담당하는 서브시스템(들)에 직접적으로 접속시키는 데이터 경로(20)이다. 패턴 데이터는 초기에 패턴 데이터 프로세싱 시스템(18)에서 준비되고, 데이터 변환 및 서브시스템들로의 스트리밍을 위해 패턴 스트리머(19)에 전송된다. 이러한 설계는, 관련 서브시스템(들)에 전달되는 패턴 데이터의 극히 높은 볼륨으로 인해서이다. 패턴 데이터는 통상적으로, 비트-맵 포맷으로 관련 서브시스템들에 스트리밍되는데, 그 이유는 데이터의 양이 서브시스템에서의 로컬 스토리지에 너무 크기 때문이다.

[0027]

오퍼레이터 인터페이스들, 그리고 상위(higer)-레벨 호스트 감독(supervisory) 및 자동화 컴퓨터들에 대한 인터페이스들은 개개의 리소그래피 엘리먼트들과 이루어지는 것이 아니라 클러스터 프론트-엔드(6)에서 이루어진다. 이는, 리소그래피 엘리먼트들(10)에 대한 인터페이스 프로토콜의 지식을 요구하는 것 없이, 그리고 부가적인 요구들을 주문하는 것 및 제어 네트워크(120) 및 데이터 네트워크(140)를 통한 통신과 간섭하는 것 없이, 이러한 인터페이스들이 클러스터를 위해 전개되는 것을 가능하게 한다.

[0028]

도 3은 리소그래피 시스템 클러스터(1)의 간략화된 평면도를 도시한다. 이러한 실시예에서, 클러스터는 5개의 리소그래피 엘리먼트들(10)의 2개의 로우들로 연이어(back-to-back) 배열된, 10개의 리소그래피 엘리먼트들(10)의 그룹을 포함한다. 클러스터(1)에 바로 인접한, 플로어(floor) 공간은 서비스 영역(23)으로서 예약된다. 각각의 리소그래피 엘리먼트는 그 자신의 진공 챔버 내에 포함된 전자-광학 컬럼을 포함하고, 이때 각각의 진공 챔버의 일 측은 기관 전달 시스템(22) 및 서비스 영역(23)을 면한다. 기관 전달 시스템(22)은 기관 공급 시스

템(24)으로부터 기판들을 수신하고, 프로세싱을 위해 상기 기판들을 리소그래피 엘리먼트들(10)에 공급하고, 그리고 프로세싱된 기판들을 리소그래피 엘리먼트들(10)로부터 수신하고, 팹(fab) 내의 다른 시스템들에 전달하기 위해, 상기 프로세싱된 기판들을 기판 공급 시스템(24)에 공급한다.

[0029] 하전 입자 리소그래피 장치의 경우에서, 진공 챔버는 바람직하게, 다수의 하전 입자 빔렛(beanlet)들을 발생시키기 위한 컴포넌트들 및 하전 입자 소스, 빔렛들을 스위칭 또는 변조하는 빔 변조 시스템, 빔렛들을 패터닝될 기판 상에 투사하기 위한 프로젝터 시스템, 및 이동가능 기판 스테이지를 둘러싼다. 진공 챔버는 바람직하게, 기판들을 기판 전달 시스템(22)으로부터 진공 챔버 내로 그리고 진공 챔버 밖으로 전달하기 위한 로드 록 시스템, 그리고 또한, 전자-광학 컬럼으로의 서비스 액세스를 위해 개방될 수 있는 서비스 영역(23)과 면하는 액세스 도어를 포함한다.

[0030] 각각의 리소그래피 엘리먼트(10)는 웨이퍼들을 수신 및 프로세싱하기 위해 독립적으로 동작한다. 각각의 리소그래피 엘리먼트는, 데이터를 프로세싱하고, 리소그래피 엘리먼트의 서브시스템들 및 컴포넌트들을 동작시키기 위한 그 자신의 컴퓨터 프로세싱 시스템들을 포함한다. 각각의 리소그래피 서브시스템(16)은 바람직하게, 서브시스템의 동작들을 지시하는 명령들을 실행시키고, 서브시스템의 동작으로부터 초래되는 데이터를 수집하고, 그리고 제어 및 데이터 네트워크들과 통신하기 위해, 그 자신의 컴퓨터 프로세서 및 메모리 시스템을 갖는다. 제어 엘리먼트 유닛(12) 및 데이터 네트워크 허브(14)는 바람직하게, 그들의 기능들을 수행하기 위한 그들 자신들의 컴퓨터 프로세서 및 메모리 시스템을 각각 포함한다. 리소그래피 엘리먼트(10)를 위한 서브시스템들(16), 데이터 네트워크 허브(14), 및 제어 엘리먼트 유닛(12)을 위한 메모리 시스템들 및 컴퓨터 프로세서는, 리소그래피 엘리먼트를 위한 진공 챔버에 가까이 접근하여, 예를 들어 진공 챔버 위에 장착된 캐비넷 내에, 또는 진공 챔버 아래 베이스먼트 내에, 또는 각각의 위치 내 부분에 위치될 수 있다.

[0031] 클러스터의 리소그래피 엘리먼트들의 연이은 레이아웃은, 제한된 풋프린트를 갖는 시스템을 제공하고, 진공 챔버들 바로 위 또는 아래의 컴퓨터 프로세서 및 메모리 시스템들의 배치는 또한 풋프린트를 감소시킨다. 팹 내 플로어 공간은 귀중하며, 따라서 팹 플로어 공간의 유용한 이용이 중요하다.

[0032] 도 4는 하전 입자 리소그래피 엘리먼트의 전자-광학 컬럼의 간략화된 개략도를 도시한다. 이러한 리소그래피 시스템들은 예를 들어, 미국 특허 번호들 제 6,897,458호; 제 6,958,804호; 제 7,019,908호; 제 7,084,414호; 및 제 7,129,502호, 미국 특허 공보 제 2007/0064213호, 및 공동-계류중인 미국 특허 출원 번호들 제 61/031,573호; 제 61/031,594호; 제 61/045,243호; 제 61/055,839호; 제 61/058,596호; 및 제 61/101,682호에 기술되며, 이들은 모두 본 발명의 소유자에게 양도되었으며, 그들 전체들이 인용에 의해 본원에 모두 포함되어 있다.

[0033] 도 4에 도시된 실시예에서, 리소그래피 엘리먼트 컬럼은, 콜리메이터 렌즈 시스템(113)에 의해 콜리메이팅되는 확대 전자 빔(130)을 생성하는 전자 소스(110)를 포함한다. 콜리메이팅된 전자 빔은 애퍼처 어레이(114a) 상에 부딪치고(impinge), 상기 애퍼처 어레이(114a)는, 서브-빔들을 포커싱하는 콘덴서 렌즈 어레이(116)를 통과하는 복수의 서브-빔들(134)을 생성하기 위해 빔의 일부를 블록킹한다. 서브-빔들은 제 2 애퍼처 어레이(114b) 상에 부딪치고, 상기 제 2 애퍼처 어레이(114b)는 각각의 서브-빔(134)으로부터 복수의 빔렛들(133)을 생성한다. 시스템은 매우 많은 수의 빔렛들(133), 바람직하게는 약 10,000 내지 1,000,000 빔렛들을 발생시킨다.

[0034] 복수의 블랭킹 전극들을 포함하는 빔렛 블랭커 어레이(117)는 빔렛들 중 선택된 것들을 편향시킨다. 편향되지 않은 빔렛들은 빔 정지 어레이(118)에 도달하고, 대응하는 애퍼처를 통과하는 반면, 편향된 빔렛들은 대응하는 애퍼처를 빗맞히고(miss), 빔 정지 어레이에 의해 정지된다. 따라서, 빔렛 블랭커 어레이(117) 및 빔 정지부(118)는 함께 개개의 빔렛들을 스위치온 및 스위치오프하도록 동작한다. 편향되지 않은 빔렛들은 빔 정지 어레이(119)를 통과하고, 그리고 타겟 또는 기판(121)의 표면에 걸쳐 빔렛들을 스캔하기 위해 빔렛들을 편향시키는 빔 편향기 어레이(119)를 통과한다. 다음으로, 빔렛들은 투사 렌즈 어레이들(120)을 통과하고, 그리고 기판을 운반하기 위한 이동가능 스테이지 상에 위치되는 기판(121) 상으로 투사된다. 리소그래피 애플리케이션들에 있어서, 기판은 일반적으로 하전-입자 감지 층 또는 저항 층을 갖는 웨이퍼를 포함한다.

[0035] 리소그래피 엘리먼트 컬럼은 진공 환경에서 동작한다. 진공은, 하전 입자 빔들에 의해 이온화되어 소스에 끌어 당겨질 수 있고, 해리되어 머신 컴포넌트들 상으로 증착될 수 있고, 그리고 하전 입자 빔들을 분산시킬 수 있는 입자들을 제거하기 위해 바람직하다. 적어도 10^{-6} 바(bar)의 진공이 통상적으로 요구된다. 진공 환경을 유지하기 위해, 하전 입자 리소그래피 시스템은 진공 챔버 내에 위치된다. 리소그래피 엘리먼트의 주요 엘리먼트들 모두는 바람직하게, 하전 입자 소스, 빔렛들을 기판 상에 투사하기 위한 프로젝터 시스템, 및 이동가능 스테이

지를 포함하는 공통 전공 챔버 내에 하우징된다.

[0036] 서브시스템들(Subsystems)

도 1 및 도 2의 실시예들에서, 각각의 리소그래피 엘리먼트(10)는 웨이퍼들을 노출시키기 위한 독립적으로 동작하는 리소그래피 엘리먼트를 형성한다. 각각의 리소그래피 엘리먼트(10)는, 엘리먼트를 위해 요구되는 기능들을 수행하도록 동작하는 다수의 서브시스템들(16)을 포함한다. 각각의 서브시스템은 특정 기능을 수행한다. 통상의 서브시스템들(16)은 예를 들어, 웨이퍼 로드 서브시스템(WLS), 웨이퍼 포지셔닝 서브시스템(WPS), 전자빔렛들을 발생시키기 위한 조명 옵틱(optic)들 서브시스템(ILO), 빔 스위칭 데이터를 리소그래피 엘리먼트에 스트리밍하기 위한 패턴 스트리밍 서브시스템(PSS), 전자빔렛들을 스위칭온 및 스위칭오프하기 위한 빔 스위칭 서브시스템(BSS), 빔렛들을 웨이퍼 상으로 투사하기 위한 투사 옵틱들 서브시스템(POS), 빔 측정 서브시스템(BMS), 계측(metrology) 서브시스템(MES) 등을 포함한다. 각각의 리소그래피 엘리먼트(10)는 바람직하게, 웨이퍼들을 수신하고, 각각의 웨이퍼를 척(chuk)에 클램핑하고, 그리고 노출을 위해 클램핑된 웨이퍼를 준비하고, 클램핑된 웨이퍼를 노출시키고, 그리고 노출된 웨이퍼를 척으로부터 언클램핑하고, 그리고 엘리먼트로부터의 제거를 위해, 노출된 웨이퍼를 제공하기 위해 구성된다.

[0038] 각각의 서브시스템(16)은, 특정 서브-기능들에 전용되고, 그리고 바람직하게 대체가능 컴포넌트들로서 설계되는 하나 또는 그 초과의 모듈들(17)을 포함할 수 있다. 모듈들(17)은 액추에이터들(19) 및 센서들(21)을 포함할 수 있고, 상기 액추에이터들(19)은 명령을 수행할 수 있는 액추에이터들(19)이고, 상기 센서들(21)은 명령을 수행하는 동안, 수행 이전 및/또는 수행 이후 동작들 및 결과들을 검출하고 측정들을 취할 수 있는 센서들(21)이다. 이러한 모듈들의 예들은, 노출 동안 기판의 이동을 위한 스테이지, 스테이지의 제어를 위한 제어 컴퓨터, 빔렛들을 기판 상으로 투사하기 위한 렌즈 어레이들에 전압들을 공급하는 투사 렌즈 모듈, 진공 챔버 내에서 진공을 발생시키기 위한 진공 펌프 등을 포함한다.

[0039] 각각의 서브시스템(16)은 독립적으로 동작하고, 그리고 인스트럭션들을 저장하기 위한 메모리 및 인스트럭션들을 실행하기 위한 컴퓨터 프로세서를 포함한다. 메모리 및 프로세서는 각각의 서브시스템 내에서 플러그-인 클라이언트(PIC)(15)로서 구현될 수 있다. 서브시스템의 적합한 구현은, 예를 들어, 리눅스 운영 체제를 실행시키는 퍼스널 컴퓨터를 포함할 수 있다. 서브시스템들은 자신들의 운영 체제를 저장하기 위한 하드 디스크 또는 비-휘발성 메모리를 포함할 수 있어서, 각각의 서브시스템들은 이러한 디스크 또는 메모리로부터 부팅한다. 이들 그리고 아래에서 논의되는 다른 피처들은, 각각의 서브시스템이, 다른 서브시스템들에 의해 부과된 제약들을 고려할 필요가 없는 독립형 유닛으로서 설계, 벨트, 및 테스트될 수 있는 자주적인(autonomous) 유닛인 설계를 가능하게 한다. 예를 들어, 각각의 서브시스템은, 다른 서브시스템들에 의해 이루어지는 메모리 및 프로세싱 용량에 대한 요구들을 고려할 필요 없이, 자신의 동작 사이클 동안 서브시스템의 기능들을 적절하게 수행할 충분한 메모리 및 프로세싱 용량으로 설계될 수 있다. 이는 특히, 이들 요건들이 유동적(in flux)일 때, 시스템의 전개 및 업그레이드 동안 유리하다. 이러한 설계의 단점들은, 총 요구되는 메모리 및 프로세싱 용량이 증가되고, 이들 컴포넌트들의 리던던시가 각각의 서브시스템 내에서 구현되어야 한다는 것이다. 그러나, 이들 단점들은, 더 신속한 발달 및 더 단순한 업그레이드를 초래하는 단순화된 설계에 의해 중요해진다(outweigh).

[0040] 서브시스템들(16)은 제어 네트워크(120)를 통해 명령들을 수신하도록 그리고 다른 서브시스템들과 독립적으로 명령들을 실행하도록 설계되어, 명령 실행에 대한 결과들을 보고하고, 요청에 따라 임의의 결과적인 실행 데이터를 전달한다.

[0041] 서브시스템들(16)은 자주적인 유닛들로서 설계될 수 있지만, 예를 들어 데이터 네트워크 허브 상의 중앙 디스크 또는 메모리로부터 부팅하도록 설계될 수 있다. 이는, 신뢰도 문제 및 각각의 서브시스템 내 개개의 하드 디스크들 또는 비-휘발성 메모리의 비용을 감소시키고, 그리고 중앙 위치에서 서브시스템을 위해 부트 이미지를 업데이트함으로써 서브시스템의 더 용이한 소프트웨어 업그레이드를 허용한다.

[0042] 통신 프로토콜들(Communication Protocols)

[0043] 서브시스템들(16)은 제어 네트워크를 통해, 지원 서브시스템 제어(Support Subsystem Control) 또는 SUCS로 또한 지정되는 엘리먼트 제어 유닛(12)에 접속된다. 엘리먼트 제어 유닛(12)은 리소그래피 엘리먼트(10)를 위해 서브시스템들의 동작을 제어하기 위한 컴퓨터 프로세서 및 메모리를 포함한다.

[0044] 클러스터 프론트-엔드(6)와 SUCS(12) 사이의 통신(102)은, PP들의 SUCS(12)로의 전달을 위해 설계된다. 자바스크립트 객체 표기법(JSON)에 기초하는 프로토콜이 이용될 수 있다. JSON은, 단순한 데이터 구조들 및 어레이들을 표시하기 위해 이용되는 인간-판독가능 데이터 교환을 위해 설계된 텍스트-기반 개방 표준이고, 그리고 구조

화된 데이터를 네트워크 접속을 통해 직렬화(serialize)하고 그리고 전송하기에 적합하다. 프로토콜에 대한 액세스는 바람직하게, 클린-룸 환경 밖의 사용자들이 아니라, 클러스터에 있는 사용자들로 제한된다. 프로토콜은 바람직하게, PJ들의 생성을 위한 인스트럭션을 제공하여, PP에 기초하여 PJ를 생성하도록 SUSC(12)에 인스트럭팅하기 위해, PP 파일 및 임의의 연관된 파라미터들을 전송한다. 부가적인 명령들은 중단(Abort) 및 취소(Cancel) 인스트럭션들을 포함할 수 있다. SUSC(12)로부터 클러스터 프론트-엔드(6)로의 통신은 확인응답 메시지들, 프로그레스 보고, 그리고 에러 및 경보 메시지들을 포함할 수 있다.

[0045] 제어 네트워크(120)를 통한 SUSC(12)와 리소그래피 서브시스템들(16) 사이의 통신(101)은 바람직하게, 네트워크 내에서의 준 실시간 수행을 보장하기 위해 단지 엘리먼트 제어 유닛 프로토콜만을 이용하여 엄격하게 제어된다. 도 5는 OSI(Open System Interconnection) 계층화된 모델에서의 엘리먼트 제어 유닛 프로토콜의 일 실시예를 도시한다. 엘리먼트 제어 유닛 프로토콜은 아래에서 더 상세하게 기술된다.

[0046] SUSD(14)와 클러스터 프론트-엔드(6) 사이의 통신(105)은 SUSD(14)로부터의 데이터 로깅, 잡 트레이싱, 및 PJ 결과들의 검색을 위해 설계된다. HTTP(Hyper-Text Transfer Protocol)이 이러한 통신 링크를 위해 이용될 수 있다.

[0047] 리소그래피 서브시스템들(16)과 SUSD(14) 사이의 통신(103)은 서브시스템들(16)로부터의 데이터의 일방향(one-way) 수집을 위해 설계된다. 데이터는 다양한 프로토콜들, 이를 테면 시스로그(syslog), HDF5, UDP 등을 이용하여 통신될 수 있다.

[0048] 고볼륨 데이터는, 핸드셰이킹(handshaking), 에러 검사, 및 정정의 큰 오버헤드 없이 데이터를 전송하기 위해 UDP(User Datagram Protocol)를 이용하여 전송될 수 있다. 결과적인 매우 낮은 송신 오버헤드로 인해, 데이터는 따라서 실시간으로 수신되는 것으로서 간주될 수 있다.

[0049] 계층적 데이터 포맷(HDF5)은 고주파 데이터의 송신 및 저장을 위해 이용될 수 있다. HDF5는 많은 양들의 수치 데이터를 저장 및 조직하는데 아주 적합하지만, 일반적으로 UDP 환경에서는 이용되지 않는다. CSV 또는 TCP와 같은 다른 데이터 포맷들은, 특히 저레벨(저볼륨) 데이터를 위해 또한 이용될 수 있다.

프로세스 프로그램(Process Program)

[0051] 리소그래피 엘리먼트(10)의 동작은, 수행될 동작들의 시퀀스를 포함하는 프로세스 프로그램(PP)(11)을 이용하여 제어된다. 엘리먼트 제어 유닛(12)에는 PP가 로딩되고, 오퍼레이터 콘솔(4)을 통해 오퍼레이터 또는 호스트 시스템(2)에 의해 요청되는 바와 같이 PP(11)를 스케줄링하고 실행시킨다. PP(11)는, 예를 들어 SEMI E40 표준에서 규정된 바와 같이, 레시피(recipe)의 역할을 할 수 있다. SEMI 표준들이, 레시피들을 처리하는 방법에 대한 많은 요건들을 명시하지만, 표준들은 모순적(contradictory)일 수 있어서, 레시피들은 바람직하게 회피된다. 대신에, 편집가능하고 포맷화되지 않은 PP가, 이른바 BLOB(Binary Large Object)들의 형태로 이용된다.

[0052] PP는, 실행(run)들 또는 프로세싱 사이클들 사이에서 변화될 수 있고, 그리고 웨이퍼의 프로세싱 환경을 결정하는 파라미터들, 설정들, 및 명령들의 세트의 사전-계획되고 그리고 재사용가능한 부분이다. PP들은 리소그래피 툴 설계자들에 의해 설계되거나 또는 툴링(tooling)에 의해 발생될 수 있다.

[0053] PP들은 사용자에 의해 리소그래피 시스템에 업로드될 수 있다. PP들은 프로세스 잡들(PJ들)을 생성하기 위해 이용된다. 프로세스 잡(PJ)은, 리소그래피 엘리먼트(10)에 의해 웨이퍼 또는 웨이퍼들의 세트에 적용될 프로세싱을 명시한다. PJ는, 명시된 웨이퍼들의 세트를 프로세싱할 때 어느 PP를 이용하는지를 규정하고, 그리고 PP로부터의 (그리고 선택적으로는 사용자로부터의) 파라미터들을 포함할 수 있다. PJ는 사용자 또는 호스트 시스템에 의해 시작된 시스템 활성화이다.

[0054] PP들은 웨이퍼들의 프로세싱을 제어하기 위해서뿐만 아니라, 서비스 동작들, 캘리브레이션 기능들, 리소그래피 엘리먼트 테스팅, 엘리먼트 셋팅들 수정, 소프트웨어 업데이팅 및 업그레이딩 등을 위해 이용될 수 있다. 바람직하게, 특정 허용된 부가적인 카테고리들 이외에, 그들이 PJ 실행에 영향을 미치지 않는 한, 서브시스템의 주기적이고 비조건적인 거동, 모듈 또는 서브시스템의 파워-업 동안의 자동 초기화, 그리고 예상되지 않은 파워-오프, 이며전시, 또는 EMO 활성화에 대한 응답같은, PP 내에 규정된 것 이외의 어떠한 서브시스템 거동도 발생하지 않는다.

[0055] PP는 단계들로 분할된다. 대부분의 단계들은 명령을 포함하고, 그리고 명령을 수행할 서브시스템을 식별한다. 단계는 또한, 명령을 수행하는데 있어서 이용될 파라미터들, 및 파라미터 제약들을 포함할 수 있다. PP는 또한, 단계가 수행될 때를, 예를 들어 동시에, 차례로, 또는 동기화되어 수행될지를 표시하기 위해, 스케줄링

파라미터들을 포함할 수 있다.

[0056] PJ의 명령 단계를 실행시키기 위해, 엘리먼트 제어 유닛(12)은 PJ 내에 표시된 명령을, PJ의 관련 단계에서 표시된 서브시스템에 전송한다. 엘리먼트 제어 유닛(12)은 타이밍을 모니터링하고, 그리고 서브시스템으로부터 결과들을 수신한다. 특정 명령들의 실행의 예들은 도 6 내지 도 13에서 예시되고, 아래에서 기술된다.

플러그-인 명령들 컨셉(Plug-in commands concept)

[0058] 엘리먼트 제어 유닛(SUSC)(12)은 (논리적 플랜을 표시하는) PP를 서브시스템 동작들의 스케줄로 변환한다. 시스템은, SUSC(12)가, 어느 서브시스템들(16)이 인스톨될지, 그리고 어느 서브시스템 명령들이 존재하는지 및 그들의 특성들이 무엇인지의 선형적인(a-priori) 지식을 갖지 않도록 설계될 수 있다. 이러한 경우에, 이러한 정보는 시동 동안 서브시스템들(16)에 의한 실행시간(runtime)에 SUSC(12)에 제공된다.

[0059] 각각의 서브시스템(16)은, 서브시스템이 파워업되고 초기화될 때, 자신들의 존재 및 능력을 SUSC(12)에 보고하도록 설계될 수 있다. 서브시스템은 제어 네트워크(120)를 통해 통신을 설정하고, 그리고 자신의 아이덴티티 및 상태를 SUSC(12)에 보고하고, 그리고 자신의 능력들, 이를 테면 어느 명령들을 자신이 수행할 수 있는지를 또한 보고할 수 있다. SUSC(12)는 PP(11)의 실행 전에 또는 실행 동안, PP를 위해 요구되는 각각의 서브시스템이, 자신의 존재 및 준비 상태를 보고했는지의 검사를 수행할 수 있고, 그리고 또한, 각각의 서브시스템이 PP 내 서브시스템에 대해 요구되는 명령들을 수행할 자신의 능력을 보고했는지를 검사할 수 있다.

[0060] 서브시스템들에 의한 이러한 셀프-보고(self-reporting)는, 리소그래피 시스템이, 단지 서브시스템들에 로컬화 이정된 소프트웨어 업그레이드들만을 이용하여 새로운 기능성으로, 또는 심지어 어떠한 소프트웨어 업그레이드도 전혀 없이 확장되거나 또는 업그레이드되는 것을 가능하게 한다. 예를 들어, 특정 서브시스템(16)은 자신의 업그레이드된 소프트웨어를 가질 수 있어서, 상기 특정 서브시스템(16)은 새로운 명령을 실행할 수 있다. SUSC(12)에는 이제, 업그레이드된 서브시스템에 대한 새로운 명령을 포함하는 새로운 PP(11)가 로딩될 수 있다. 업그레이드된 서브시스템(16)이 파워업될 때, 상기 업그레이드된 서브시스템(16)은 엘리먼트 제어 유닛(12)과 통신하여, 자신의 아이덴티티 및 상태, 그리고 새로운 명령을 포함하여 어느 명령들을 자신이 실행할 수 있는지를 표시한다. SUSC(12)는, 새로운 명령을 실행하기 위해 서브시스템에 명령하는 단계를 포함하는 PJ를 발생시키도록 PP를 스케줄링한다. SUSC(12)는 업그레이드된 서브시스템이, 자신이 새로운 명령을 수행할 수 있다는 것을 보고했는지의 검사를 수행할 수 있다. 이러한 업그레이드는 따라서, SUSC(12) 또는 다른 서브시스템들(16) 중 임의의 서브시스템이 아닌, 단지 관련 서브시스템의 그리고 PP의 업그레이드만을 요구한다.

제어 네트워크(Control Network)

[0062] 제어 네트워크(120)는 바람직하게, 실시간 통신 프로토콜을 이용하지 않지만, 그럼에도 불구하고 준 실시간 수행을 제공하도록 설계되어서, 동일한 타이밍 거동을 초래하는 동일한 조건들 하에서 동일한 잡의 실행을 제공하기 위해, 높은 반복성 – 예를 들어, 실질적으로 1 밀리초 이내의 반복성 – 을 갖는 SUSC(12)와 서브시스템들 사이의 통신을 제공한다.

[0063] 제어 네트워크(120)를 통한 이러한 준 실시간 수행은 아래의 조치들 중 몇몇 또는 모두를 구현함으로써 달성될 수 있다.

[0064] (a) 리소그래피 시스템은 PJ의 실행을 시작하기 전에 PJ를 발생시키도록 각각의 PP를 스케줄링하여, PJ의 각각의 단계를 시작할 시간(그리고 또한, 완료할 시간을 포함할 수 있음)을 결정한다. 완전한 PJ의 그리고 각각의 단계에 대한 스케줄링된 시작 시간(및 완료 시간)은 PJ가 개시되기 전에 완전히 결정될 수 있고, 그리고 이들 시간들은 클러스터 프론트-엔드에 보고될 수 있다.

[0065] (b) PP(및 PJ)는 어떠한 조건적인 단계들 또는 동작들도 없이, 그리고 어떠한 재시도들도 없이 설계될 수 있다. 리소그래피 엘리먼트 내에서의 동작들이 동시에 수행될 수 있더라도, 예를 들어 명령들이 상이한 서브시스템들 상에서 동시에 실행되더라도, PP/PJ의 단계들은 동작들의 시퀀스를 기술한다. 조건적인 단계 또는 동작은, 수행할 2개의(또는 그 초과의) 대안적인 단계들을 규정함으로써 PP 내에 프로그래밍될 수 있으며, 여기서 대안적인 단계들은 별별 경로들로서 배열된다. 대안적인 단계들은, 동일한 실행 시간이 각각 할당된 별별 경로들로서 PJ 내에서 스케줄링되어서, 전체로서 PJ의 실행 시간은, 어느 대안적인 경로가 서브시스템들 상에서의 실행을 위해 선택되는지에 따라 변화하지 않는다.

[0066] PJ의 단계들의 실행 시간은 선행 또는 별별 단계의 결과들에 조건적이지 않지만, 미리결정된 스케줄에 따라 실행된다. 단계가 정확하게 또는 스케줄링된 시간 내에 완료되지 않는다면, 다시(again) 실행되지 않을 것이지만

(즉, 어떠한 재시도들도 존재하지 않음), 대신에 다음(next) 스케줄링된 단계가 실행될 것이다.

[0067] (c) 스케줄링 후에, 모든 각각의 단계는 자신의 미리결정된 스케줄링된 시간에, 바람직하게는 약 1 밀리초의 타이밍 정확도 내에서 SUSC(12)에 의해 실행될 것이다. 스케줄링된 시간에서, SUSC(12)는 관련 단계를 실행하고, 서브시스템에 의한 실행을 위해 그러한 단계에 대한 임의의 명령을 관련 서브시스템(16)에 전송할 것이다. SUSC(12) 내 스케줄링된 타이밍은, 선행 단계로부터의 명령을 실행하기 위해 성공적인 완료 또는 심지어 실패에 관하여 서브시스템들(16)로부터의 피드백에 따르지 않는다. 단계가 실행 시간 기간을 규정하는 경우, SUSC(12)는 PJ의 다음 단계로의 진행 전에 그러한 시간 기간이 경과하는 것을 대기한다. 서브시스템이 이러한 기간의 만료 전에 명령의 실행으로부터의 결과를 리턴하는 경우, SUSC(12)는 다음 단계로의 진행 전에 시간 기간의 만료를 여전히 대기할 것이다. 서브시스템이 여러 메시지를 리턴하는 것을 실패하는 경우, SUSC(12)는 그럼에도 불구하고 시간 기간의 만료를 대기할 것이고, 그 다음으로 다음 단계로 진행할 것이다. 서브시스템이 명령을 적합하게 실행하는 것을 실패하는 경우, 오퍼레이터 또는 호스트 시스템은, 어떠한 시정 동작(corrective action)이 필요할 수 있을지를 결정할 것이다.

[0068] 단계의 실행에 있어서의 임의의 타이밍 변화가 스케줄링 시에 고려되어서, 단계의 실행을 위해 스케줄링된 시간은 가장 큰 예상되는 실행 시간보다 더 크다. 시간 스케줄은 통상적으로 PP에서 규정된다. 스케줄링 시기(moment)는, 실행을 시작하기 전의 적기(just-in-time), 또는 실행 전의 어떠한 시간, 예를 들어 단계를 실행 큐에 부가할 때 일 수 있다.

[0069] (d) 서브시스템들(16)에 의해 전송 및 실행되는 명령들은 고정된 최대 시간을 갖는 명령의 실행을 명시한다. 가장 큰 실행 시간은, 자신의 상황들이 주어지면 서브시스템에 의해 결정되는 바와 같은 미리-결정된 최대 기간이다. 오퍼레이터 또는 호스트 시스템은, 이러한 시간이 초과되는 경우 어떠한 시정 동작이 필요할 수 있는지를 결정할 것이다.

[0070] (e) 서브시스템이 수행할 명령, 및 관련 서브시스템에 전송될 명령과 연관된 데이터를 갖는 PJ 단계를 실행할 때, 데이터는, 명령 그 자체가 서브시스템에 전송되는 것보다 미리 서브시스템에 전송될 수 있다. 데이터는 하나 또는 그 초과의 메시지들에서 전송되어서, 임의의 하나의 메시지는 크기 제한된다. 데이터는, 상기 데이터가 서브시스템에 의해 수신되는 것을 보장하기 위해 충분히 미리 전송된다, 즉 데이터를 전송하는 것과 명령을 전송하는 것 사이의 시간은 예상된 송신 시간보다 매우 더 크고, 그리고 서브시스템은 데이터의 수신을 확인응답할 수 있다. 이는, 타이밍이 중요하지 않은 경우, 데이터를 포함하는 더 큰 메시지들이 미리 전송되는 것을 가능하게 하고, (연관된 데이터 없이) 단지 명령만을 포함하는 매우 더 작은 메시지는 스케줄링된 시간에서 전송될 수 있다. 이러한 조치는, 송신 시간을 감소시키고, 그리고 명령의 적시의 그리고 신뢰적인 송신을 증가시키기 위해, 네트워크 상의 부하를 분산시켜, 명령이 전송될 때 혼잡을 회피시킨다.

[0071] (f) 유사하게, 서브시스템이 명령의 실행을 완료할 때, 상기 서브시스템은 명령이 완료되었다는 것을 표시하는 짧은 메시지를 SUSC(12)에 전송할 수 있지만, 나중 시간에서의 SUSC(12)에 의한 검색을 위해 명령의 실행으로부터 초래된 임의의 데이터를 훌딩한다. SUSC(12)가 명령 완료의 표시를 수신하는 경우, 그 다음으로 상기 SUSC(12)는 결과 데이터를 검색하라는 요청을 서브시스템에 전송할 수 있다. 이는, 송신 시간을 감소시키고, 그리고 적시의 그리고 신뢰적인 송신을 증가시키기 위해, 네트워크 상의 부하를 분산시켜서, 명령 완료 표시가 전송될 때 혼잡을 회피하고, 그리고 그 다음으로 데이터는, 후에 타이밍이 중요하지 않을 때 검색될 수 있다.

[0072] (g) 서브시스템들(16)과 SUSC(12) 사이의 메시지 교환은, SUSC(12)로부터의 인스트럭션들의 수신을 확인응답하기 위해 리플라이 메시지들을 포함한다. 리플라이 메시지들은, 서브시스템(16)이 관련 메시지를 SUSC(12)로부터 수신하였고 여전히 기능하고 있다는 확인 이외의 어떠한 다른 정보도 반송하지 않는다. 이는, SUSC(12)가, 예상되는 바와 같이 서브시스템(16)이 응답하는 것을 중지하는 때를 검출하는 것을 가능하게 하여서, 이는 시기 적절하게 클러스터 프론트-엔드에 보고될 수 있다. 단순한 소프트웨어 구현을 유지하기 위해, 서브시스템(16)에 대한 어떠한 타임아웃들도 존재하지 않는다. 리플라이 메시지들에 대해 예상되는 총 응답 시간은 통상적으로 짧으며, 예를 들어 왕복 0.5ms 미만이다. 비-응답 서브시스템(16)을 검출하기 위해, SUSC(12)는 예상되는 응답 시간을 훨씬 초과하는 타임아웃을 가지며, 예를 들어, 응답이 예상되는 이후 30 초이다. 이러한 타임아웃은, 서브시스템(16)이 기능하지 않는다는 합리적인 의심의 여지없다는 결론을 내리기에 충분히 크지만, 오퍼레이터(4)가 정상적으로 그렇게 하기 전에 실패를 검출하기에 충분히 작아서, 오퍼레이터(4)는 무엇이 진행중인지 (what is going on) 궁금해 할 필요가 없다.

[0073] (h) 제어 네트워크(120)는 한번에(at a time) 데이터 패키지, 예를 들어 단일 메시지 시퀀스의 단일 전달에 안전한 프로토콜 제어기를 포함한다.

- [0074] (i) 제어 네트워크는 TCP 링크 상에서 단순 송신 프로토콜, 예를 들어 바이트 프로토콜을 이용한다. 복잡한 또는 예측불가한 엘리먼트들을 포함하는 송신 프로토콜들은 회피된다. 제어 네트워크 상에서 이용되는 프로토콜은, 다수의 데이터 전달들이 동시에 발생할 수 없다는 것을 보장하여서, 네트워크 상에서 예측불가한 이벤트들은 발생하지 않는다.
- [0075] TCP 접속은 2 바이트 스트림들로 구성되고, 각각의 방향으로 1 바이트이다. 프로토콜은 이를 스트림들 양측 모두를 별개의 메시지들로 분리하고, 각각의 메시지는 0 또는 그 초과의 바이트들의 임의의 데이터의 시퀀스이고, 시퀀스의 길이를 표시하는 4 바이트들이 선행한다. 이러한 길이는 무부호 정수로서, 예를 들어 네트워크 순서 ("빅-엔디언(big-endian)")로 32비트 정수로서 인코딩될 수 있어서, (232-1) 바이트들의 최대 메시지 크기를 제공한다. 이러한 메시지 분리 메커니즘은 피تون(Python) 프로그래밍 언어로 "접속 객체들(Connection Objects)"에 의해 구현될 수 있다. 표준 멀티프로세싱 접속 패키지를 이용함으로써, 피تون 그 자체는 프로토콜의 이러한 부분을 핸들링할 수 있다. 다른 프로그래밍 환경들에서는, 이러한 양상을 정확하게 구현할 필요가 있을 것이다. 이들이 규정된 시퀀스들을 따른다면, 메시지들은 어느 한 측에 의해 개시될 수 있다.
- [0076] 메시지 시퀀스의 모든 각각의 제 1 메시지는 자바스크립트 객체 표기법(JSON)으로 인코딩된 제어 데이터 구조를 포함할 수 있다. SUSC 프로토콜 JSON 메시지들 모두는 동일한 일반적인 구조를 가질 수 있으며; 2개의 엘리먼트들의 어레이, 첫 번째는 메시지 타입을 갖는 스트링을 포함하고, 두 번째는 그러한 메시지에 대한 네임드 아규먼트들을 갖는 사전을 포함하며, 예를 들어 다음과 같다 ["<messageType>", {"param1" : <value1>, "param2" : <value2>, ...}].
- [0077] 몇몇 메시지 시퀀스들, 예를 들어 인터페이스 명령들의 입력 및 출력 아이템들을 전달하기 위해 이용되는 것들에서, 이러한 제어/명령 메시지에 이어 여러 데이터 메시지들이 바로 뒤따르며, 예를 들어 입력/출력 아이템 당 (per) 하나의 데이터 메시지이다. 프로토콜은 이를 메시지들에서 이용되는 인코딩에 관한 어떠한 추정들도 하지 않으며, 대신에 해석은, 서브시스템 명령들, 데이터 및 거동을 명시하는 설계 문서 내에 기술되고, 예를 들어 "list of tuples of 2 floats, encoded with pickle", 또는 "pressure curve plot as image/pgn"이다.
- [0078] 임의의 메시지들을 JSON 제어 메시지들과 믹싱하는 이러한 방식은, 프로세스 잡 출력들이, CSV, PDF, PNG 같이, 오퍼레이터에 의해 이해가능한 데이터 포맷으로 플러그-인 소프트웨어에 의해 직접적으로 생성되는 것을 허용한다. 이는 또한, 이러한 객체들의 송신이, 예를 들어 리눅스의 센드파일(sendfile) 기능을 이용함으로써, 데이터 스트림의 부분으로서의 이스케이핑(escaping) 또는 부가적인 인코딩 없이 수행될 수 있다는 것을 의미한다. 이러한 자유(freedom)는 또한, 특정 인터페이스 기능을 위해 더 적합한 인코딩에 대해 서브시스템들 사이에서 구체적으로 동의하기 위해 이용될 수 있다. 그러나, 모든 각각의 서브시스템이 그 자신의 데이터 인코딩 방식을 갖는 경우에 이는 혼명하지 못할 것이며, 따라서 서브시스템이 어떠한 특정한 요건들도 갖지 않는 경우, 요건들 사이의 최상의 트레이드-오프를 구현하기 위해, 디폴트 방식이 선택되어야 한다.
- [0079] (j) PIC(15) 설계를 단순화하기 위해, 서브시스템들은, SUSC 프로토콜 예외들을 검출하고, 그리고 SUSC(12)의 정확한 거동을 검증하는 것을 담당하지 않는다. SUSC 프로토콜의 목적은, 사용자에 의해 생성된 프로세스 잡들 내 프로세스 프로그램들로부터의 프로세스 단계들을 실행하는 것이다. 예외들은, 예상되지 않은 거동을 사용자에게 통신하기 위해 이용된다. 물론 명령 실행에 있어서의 예외들 외에도 또한, SUSC 프로토콜에 따르지 않음으로써 예외들이 발생할 수 있다. SUSC(12)는 예외들을 로깅(log)하는 것을 담당하며, PIC가 SUSC 프로토콜에 따라 거동하지 않을 때(또는 전체적으로 응답하는 것을 중지했을 때) PIC에 대한 접속을 클로징할 수 있지만, 그러한 PIC는 단순한 PIC 요건들을 유지하기 위해 SUSC의 정확한 거동을 검증하는 것을 담당하지 않는다. PIC에 의해 프로토콜 위배(violation)에 직면하는 경우, 이는, 상기 프로토콜 위배에 적절하게 응답할 수 있거나 또는 응답하지 않을 수 있으며, 그리고 후속하는 메시지들이 이해되어야 한다는 어떠한 요건도 존재하지 않는다. 몇몇 진단들을 로깅하고, 예외를 로깅하고, 메시지가 핸들링될 수 없을 때 접속해제 및 재접속하도록 시도하는 것이 권고된다. 이는 요건이 아니라 권고인데, 그 이유는 이러한 재접속 후에 PIC가 다시 이용가능 상태에 있게 된다는 어떠한 보장 또는 기대도 존재할 수 없기 때문이다(그러나, 적어도 이는 거의 이해가능한 상태에 있을 것이다).
- [0080] (k) 오퍼레이터들 및 상위-레벨 자동화 또는 감독 웹 컴퓨터 시스템들을 위한 인터페이스들은 제어 네트워크(120) 상의 통신들로부터 제거된다. 이를 인터페이스들은 클러스터 프론트-엔드(6)로부터 제공되어서, 이들은 제어 네트워크(120) 상에서 혼잡을 발생시키지 않고, 서브시스템들(16)과의 통신들의 타이밍에 영향을 미치지 않는다.
- [0081] (l) 데이터 수집은, 제어 네트워크(120) 상의 제어 통신들로부터 별개의 데이터 네트워크(140)로 완전히 분리된

다. 이러한 기능들의 분리는 아래에서 더욱 상세하게 논의된다.

[0082] 동기 클록(Sync Clock)

리소그래피 시스템은, 서브시스템들에 의한 그리고 시스템 내에서의 동작들의 동기화를 가능하게 하기 위해 제어 네트워크 상에서 클록 신호를 제공한다. 시스템은, 상이한 타이밍 정확도들을 위해 클록들을 서브시스템들에 제공하기 위해, 상이한 주파수들로 2개의 클록 신호들을 제공할 수 있으며, 예를 들어 하나의 클록 신호는 밀리초들 내로 정확하고, 그리고 하나는 나노초들 내로 정확하다. 예를 들어, 범 스위칭 데이터를 스트리밍하기 위한 패턴 스트리밍 서브시스템 및 전자 범랫들을 스위칭온 및 스위칭오프하기 위한 범 스위칭 서브시스템은, 범랫들의 고주파 스위칭을 가능하게 하기 위해 매우 고주파로 데이터를 교환할 필요가 있으며, 범 측정 시스템은 매우 고주파로 범 위치 측정들을 전송하도록 요구된다. 높은 정확도 클록을 필요로 하는 다른 기능들은, 기관 포지셔닝 시스템 및 투사 옵틱들 시스템을 포함한다. 이들 서브시스템들에는 동기 클록 서브시스템에 의해 제공된 나노초 클록 펄스들이 공급되고, 이들 서브시스템들은 상기 나노초 클록 펄스들을 수신 할 수 있다.

[0084] 제어 네트워크 프로토콜 메시지들(Control Network Protocol Messages)

도 6a 내지 도 6h는 프로토콜에 의해 지원되는 메시지 시퀀스들(30, 40, 50, 70, 80, 90, 100)을 도시한다. 시퀀스 도면들(30, 40, 50, 70, 80, 90, 100)에는, SUSC(12)가 접속을 클로징하기 전에 이용할 타임-아웃들에 대해 그리고 평균 경우에 대해 요구되는 응답 시간들이 주석으로 달려 있다. 기술된 타이밍들은 원칙적으로 SUSC(12), SUSD(14), 및 서브시스템(16)을 수반하는 왕복 시간들이다.

SUSC(12) 프로토콜은 TCP 접속을 통해 메시지들을 통신한다. SUSC(12)는 TCP 서버로서 동작하고, 서브시스템들(16) 내 PIC들(15)은 TCP 클라이언트로서 동작한다. PIC들은 특정 포트 상에서 SUSC(12)로의 접속을 이루고, SUSC(12)는 제어 네트워크(120) 내에서 IP 어드레스를 갖는다. 접속이 실패하는 경우, PIC는 접속이 성공할 때 까지 재시도한다. 도 6a는 엘리먼트 제어 유닛(SUSC)(12)과 서브시스템(16) 내 PIC가 서로 접속하는 접속 시퀀스(30)를 도시한다. SUSC(12) 및 서브시스템(16)으로부터의 수직 시간 라인들(27, 29)은 메시지들의 타이밍을 표시하기 위해 도시된다. 몇몇 시간 포인트에서, 접속 시퀀스는, 운영 체제 레벨에서 접속을 셋업하기 위해 TCP 프로토콜 또는 TCP 접속(33)의 표준 3-웨이(way) 핸드셰이크로 시작한다. TCP 접속(33)이 설정되면, 서브시스템(16)은 접속 메시지(35)를 SUSC(12)에 전송한다. SUSC(12)는 접속 리플라이(37)를 시간 기간(31), 바람직하게는 0.5초 내에 전송한다. 접속 리플라이(37)는 위치 특정 정보, 예를 들어 머신이 위치되는 타임 존(time zone)을 포함할 수 있다. 이는, SUSC(12)의 타임 존(39)을 설정하기 위해 이용될 수 있다. 서브시스템(16)은 나중에 내부 서브시스템 초기화(41)를 추가로 수행할 수 있다.

[0087] 원칙적으로, 접속은 무기한으로, 즉 측(side)들 중 하나가 셧다운되고 접속이 클로징될 때까지, 개방되어 유지된다. 이러한 클로징은 운영체계의 셧다운 시퀀스의 부분으로서 운영 체계에 의해 자동적으로 또는 애플리케이션에 의한 명시적인 소켓 클로즈에 의해 발생한다. PIC(15)가 그의 접속의 클로징을 검출할 때, 상기 PIC(15)는 재접속하도록 시도하고, 그리고 재접속이 실패하는 경우, 상기 PCI(15)는 재접속이 성공할 때까지 재시도할 수 있다. SUSC(12)가 접속의 클로징을 검출할 때, 상기 SUSC(12)는 (등록된 명령들 또는 계획된 장래의 명령들과 같은) 연관된 PIC의 모든 정보를 폐기하고, 그것이 재접속할 때까지 PIC(15) 및 연관된 서브시스템(16)은 비어있는(absent) 것으로 고려할 수 있다.

[0088] SUSC(12)가 접속의 클로징을 검출하지 않고, 그리고 이중(duplicate) 접속이 PIC(15)에 의해 설정되는 특별한 경우에, SUSC(12)는 이전의 접속이 클로징된 것으로 고려할 수 있다. 이러한 시나리오는, TCP 접속들을 클로징 할 기회가 PIC(15)의 커널에 주어지지 않은 채, PIC(15)가 갑자기 파워 오프되고, SUSC(12)가 이러한 PIC와의 통신을 아직 시도하지 않은 경우에 발생할 수 있다.

[0089] 이러한 거동은 어느 한 측이, 다른 측의 재시동을 요구함이 없이 재시동될 수 있다는 것을 보장한다. 도 6b는, 서브시스템(16)이 이미 초기화되었지만, SUSC(12)로의 자신의 접속을 상실한 경우에 시동되는 재접속 시퀀스(40)를 도시한다. 이는 SUSC(12)에 재접속하는 서브시스템(16)의 책무(responsibility)이다. 반복은, 접속 메시지(35)를 SUSC(12)에 전송하는 서브시스템(16)에 의해 수행될 것이며, 접속 메시지(35)를 수신시, SUSC(12)는 접속 리플라이(37) 메시지를 서브시스템(16)에 다시(back) 전송한다. 여기서, 이는 또한, 시간 기간(31), 바람직하게는 0.5초 이내이다. 재접속 시퀀스(40)는 서브시스템(16)이 SUSC(12)에 접속될 때까지 반복될 것이다.

[0090] 도 6c는 실행 명령 시퀀스(50)를 도시하며, 상기 실행 명령 시퀀스(50)는 선택적으로, SUSC(12)에 의한, '설정

(set)' 명령(51)을 갖는 입력 아규먼트들의 서브시스템(16)으로의 전달로 시작한다. 설정 명령(51)에 이어, N 회(times)의 서브시스템(16)에 전송되는 상당한 양의 데이터(52)가 뒤따른다. 최종 데이터(52)가 전송된 후에, 서브시스템(16)은, 설정 리플라이(53)(setReply)를 SUSC(12)에 다시(back) 전송함으로써, 수신된 명령(51) 및 데이터(52)를 확인응답한다. 서브시스템(16)으로부터 SUSC(12)에 전송된 임의의 리플라이 시간 기간(55)을 갖고, 상기 리플라이 시간 기간(55)은 바람직하게 평균 0.5밀리초 미만을 포함한다. 설정 리플라이(53)는 타임-아웃 제한(54) 내에 전송되어야 하며, 상기 타임-아웃 제한(54)은 바람직하게 30.0초 플러스 마이너스 1.0초를 포함한다. 리플라이가 수신되지 않는 경우, SUSC(12)는 서브시스템(16)이 접속되지 않았거나 결합있는 것으로 고려할 것이다. SUSC(12)가 실행 명령(57)을 전송한 후에, 서브시스템(16)은 바람직하게, 리플라이 시간 기간(55) 내에, executeStarted 리플라이(58)를 이용하여 확인응답한다. 서브시스템(16)이 실행을 완료한 후에, 상기 서브시스템(16)은 executeDone(59)를 SUSC(12)에 전송한다. (존재한다면) 출력 아규먼트들은 SUSC(12)에 의해 전송된 '획득(get)' 명령(60)을 이용하여 검색된다. 서브시스템(16)은 getReply(61)를 이용하여 확인응답하고, 그 후에, N 데이터 양(62) 또는 출력 아규먼트들을 전송한다.

[0091] 장래의 실행 명령들(57)을 위해 더 이상 필요하지 않은 입력 및 출력 아규먼트들은 삭제 명령(63)을 이용하여 서브시스템(16)으로부터 제거된다. 서브시스템(16)은 바람직하게 deleteReply(64)를 이용하여 리플라이 시간 기간(55) 내에 리플라이할 것이다. 프로토콜의 하나의 버전에서, 입력들 및 출력들은 하나의 인터페이스 명령으로부터 다른 인터페이스 명령으로 이월(carry over)되지 않는다. 서브시스템(16)은, 다음(next) 명령을 위해 필요한 것보다 적은 입력들이 전송될 것이라는 것을 안전하게 추정할 수 있고, 그리고 상기 서브시스템(16)은, 다음(next) 명령들의 입력들이 전송되기 전에, 모든 입력들 및 출력들이 삭제될 것이라는 것을 안전하게 추정할 수 있다.

[0092] 도 6d는 중단된 실행 명령(71)의 경우에서의 이벤트들의 순서를 예시하는 중단 시퀀스(70)를 도시한다. 중단 시퀀스(70)는 전역적으로(globally) 정상 실행 시퀀스(50)와 동일하며, 아래의 차이들을 갖는다. SUSC(12)는, executeStarted(58) 메시지와 executeDone(59) 메시지를 수신하는 사이의 기간 내에 중단 요청(71)을 전송한다. 서브시스템(16)은 차례로, abortReply(72)를 전송한다. 중단 요청(71)에 대한 응답으로서 abortReply(72)를 SUSC(12)에 전송하는 것은, 90.0초 플러스 또는 마이너스 1.0초를 포함하는 시간 기간 abortReply(74) 내에 수행되어야 하며, 여기서 90초는, 최대 폴링 간격을 위한 60초 플러스 부가적인 30초를 포함한다. 현재 실행중인 명령에 따라, 명령은 완료로 진행되거나 또는 명령이 끝나기 전에 취소(break off) 된다. 명령이 끝나지 않은 경우, 예외(exception)(91)가 로깅될 수 있다. 또한, 모든 출력 아규먼트들이 이용 가능한 것은 아니며, 비어있는 것들(0 바이트)은 서브시스템(16)에 의해 대체될 수 있다. SUSC(12)는 그들 모두를 검색하도록 시도할 것이다. SUSC(12)로부터의 중단 메시지(71) 및 서브시스템(16)으로부터의 executeDone(59)가 서로 교차하는 것이 가능하다. 그러한 경우, 도 6e에 도시된 시퀀스(80)가 적용가능하다.

[0093] 도 6e는 경합 조건(80)을 갖는 중단 시퀀스를 도시한다. 경합 조건(81)은, 둘 또는 그 초과의 메시지들이 서로 교차하는 경우에 발생하는 조건이다. 여기서, 중단 메시지(71)는 executeDone 메시지(59)와 교차한다.

[0094] 도 6f, 도 6g, 도 6h는 예외 시퀀스(90)에 대한 3개의 경우들을 도시한다. 예외 메시지들은, 예외(90)가 예를 들어, 에러 또는 경보의 형태로 발생하는 경우에, 서브시스템(16)에 의해 SUSC(12)에 전송된다. 에러는, 리소그래픽 엘리먼트가 예상되는 바와 같이 거동하지 않는다는 것을 사용자(4)에게 통신하기 위해 이용된다. 경보는 사용자(4)에 의해 확인응답되어야 하는 조건이다.

[0095] 인터페이스 명령을 실행하는 콘텍스트 내에서 발생하는 예외들(91)은 콘텍스트 정보로 태깅(tag)될 수 있다. 이러한 명령의 콘텍스트 밖에서 발생하는 예외들(91)에 있어서(자발적 예외들(90)), 이를 콘텍스트 필드들은 빈 채로 남겨질 수 있거나, 또는 실패와 관련되는 이미 완료된 명령의 가장 적합한 콘텍스트가 공급될 수 있다.

[0096] 특히, 자발적인 예외들(90)의 경우에서, 사용자(4)에게 예외들이 쇄도하지 않도록 주의가 이루어져야 한다. 예를 들어, 에러 조건이 고주파 제어 루프 내에서 겸출되는 경우, 이상적으로는 단지 하나의 예외(91)만이 그러한 조건에 대해 데이터 네트워크 허브(14)에 의해 로깅되고, 루프의 모든 각각의 반복(iteration)에 대해서는 반복적으로 로깅되지 않아야 한다. 또한, 조건이 계속되는(persist) 경우에, 예외(91)를 낮은(바람직하게는 1 분) 빈도로 반복하거나, 또는 조건이 변경되거나 또는 사라질 때(이는 또한 예상되지 않아서, 여전히 예외(91)로 고려할 수 있음), 팔로우-업(follow-up) 예외(91)를 로깅하는 것이 더 나을 것이다.

[0097] 복구 프로시저를 통해 오퍼레이터(4)를 가이드하고, 그리고 예를 들어, 전화 또는 트러블 티켓 타이틀에 의해 간결한 방식으로 문제의 본질을 통신하기 위해, 예외들(91)은 오퍼레이터(4)에게 여러 상황들을 통지하도록 의도된다. 그러므로, 예외들(91)은 모듈 설계자들을 위한 디버깅 정보 채널로서 고려되지 않아야 한다. 대안적

으로, 인간 판독가능 테이터 포맷들의 PJ 출력들은, 모니터링되거나 또는 오프라인으로 조사되어야 하는 정보를 위해 이용될 수 있다.

[0098] 예외 메시지들(91)은 프로토콜 시퀀스들을 종결시키기 않고, 장래의 명령들을 핸들링하는 것을 중지할 이유가 없다. 예를 들어, 인터페이스 명령이 실패하고, 예외(91)가 데이터 네트워크 허브(14)에 의해 로깅된 후에, executeDone(59)은 여전히 전송될 수 있고, 출력들은 '획득(get)'하도록 이용가능하다. 서브시스템(16)은 또한, 상기 상황으로부터 복구하기 위해 새로운 인터페이스 명령들을 수용할 준비가 되어야 한다.

[0099] 예외(91)는, getReplay(61) 시퀀스 내에서는 제외하고, 프로토콜 내에, 임의의 시간에 삽입되도록 허용된다. 임의의 예외들(91)이 병렬 쓰레드에 의해 발생될 수 있는 경우, PIC(16) 설계자는, 임의의 명령들에 대해서가 아니라, 이러한 예외들(91)을 로깅하기 위한 전용의 PIC(16) 접속을 할당하도록 고려할 수 있다. 이는 단순한 방식으로 이러한 간섭을 회피한다.

[0100] 도 6f는 SUSC(12)로부터 전송되는 포괄적인 또는 임의의 프로토콜 명령(92) 동안 서브시스템(16)에 의해 발생되는 예외(91)를 도시한다. 예외(91) 후에, 서브시스템(16)은 또한, 임의의 프로토콜 명령 리플라이(93)를 SUSC(12)에 전송한다. 도 6g는 자발적 예외(91)를 도시하고, 그리고 도 6h는 실행 명령(57) 동안 서브시스템(16)으로부터 SUSC(12)에 전송된 예외(91)를 도시한다. 임의의 프로토콜 명령(92) 동안, 예외(91)가 발생될 수 있다. 명령은 정상적으로, 연관된 리플라이 메시지를 이용하여 종결된다. 도 6g에 도시된 바와 같이, SUSC(12)로부터의 명령들 사이에, 예외(91)가 로깅되는 것이 또한 가능하다. 예를 들어, 주기적 검사로부터 또는 허브 초기화 시퀀스로부터이다. 예외(91)가 실행 명령(57) 동안 전송될 때, 어떠한 예외(91)도 발생하지 않은 것처럼, 획득(60) 및 삭제(63) 명령들이 실행된다. 그러나, 예외(91)의 발생으로 인해, 몇몇 출력 아규먼트들이 빌(empty)(0 바이트) 수 있다는 것이 가능하다.

데이터 네트워크(Data Network)

[0102] 서브시스템들(16)은 데이터 네트워크(140)를 통해 데이터 네트워크 허브(14)에 접속된다. 데이터 수집, 저장, 및 관리는 데이터 네트워크(140)를 통해 수행된다. 제어 네트워크(120)는 엘리먼트 제어 유닛(12)과 리소그래피 서브시스템들(16) 사이에 제어 네트워크 경로를 형성하고, 데이터 네트워크(140)는 데이터 네트워크 허브(14)와 리소그래피 서브시스템들(16) 사이에 데이터 네트워크 경로를 형성한다. 제어 네트워크(120) 및 데이터 네트워크(140)는 물리적으로 별개의 네트워크들이다. 각각의 네트워크는, 배선, 네트워크 컴포넌트들, 이를 테면 스위치들, 및 서브시스템들(16)로의 네트워크 접속들을 포함하는, 그 자신의 별개의 물리적 매체를 갖는다. 따라서, 제어 네트워크 경로 및 데이터 네트워크 경로는 물리적으로 별개의 매체를 포함하고, 별개의 그리고 독립적인 통신 경로들을 형성한다.

[0103] 각각의 리소그래피 서브시스템(16)은, 제어 네트워크를 통해 엘리먼트 제어 유닛(12)으로부터 그리고 엘리먼트 제어 유닛(12)에 제어 정보를 수신 및 전송하도록 적응된, 제어 네트워크(140)로의 접속을 갖는다. 각각의 리소그래피 서브시스템(16)은 데이터 네트워크를 통해 데이터 네트워크 허브(14)에 데이터 정보를 전송하도록 적응된, 데이터 네트워크(140)로의 별개의 접속을 갖는다.

[0104] 데이터 네트워크(140)는 서브시스템들(16)에 의해 이용되는 송신 레이트들보다 매우 더 큰 대역폭으로 설계된다. 예를 들어, 데이터 네트워크(140)는 1 Gbit/s의 대역폭 및 100Mbit/s로 전송하도록 설계된 서브시스템들을 가질 수 있다. 데이터 네트워크 내에서, 어떠한 네트워크 제어기 또는 데이터 전달들의 조정도 존재하지 않는다. 통상적으로 데이터 네트워크(140) 내에서, 서브시스템들(16)과 데이터 네트워크(140) 사이의 데이터 교환 캡(cap)은 140Mbit로 설정되는 반면, 데이터 네트워크(140)와 SUSD(14) 사이의 데이터 교환에 대해서는 1Gbit의 캡이 설정된다. 네트워크 스위치는, (데이터) 패키지 충돌을 방지하기 위해 트래픽을 합체(merge)하도록 데이터 네트워크(140) 내에 포함된다.

[0105] 데이터 네트워크 허브(14)는 서브시스템들(16)로부터 수신된 데이터를 지속적으로 로깅하도록 적응된다. 이러한 데이터는, PJ의 실행 동안 서브시스템들에 의해 취해진 측정 데이터, 서브시스템들의 설정들, 그리고 디버깅 및 장애 추적에 대한 데이터를 포함할 수 있다. 데이터 네트워크 허브(14)는 바람직하게, 저레벨 추적 데이터를 포함하는 모든 데이터에 항상 로깅하여, 데이터 로깅을 턴온할 필요가 없다. 이러한 지속적인 데이터 로깅은 문제 진단을 가속화(speed up)시켜서, 여러 또는 문제를 재생성 및 재실행할 필요성을 감소시킨다.

[0106] 엘리먼트 제어 유닛(12)은, 데이터 네트워크 허브(14)에 의한 로깅을 위해 엘리먼트 제어 유닛(12) 내에서 실행되는 PJ들의 진행에 관한 정보를 전송하기 위해 데이터 네트워크 허브(14)에 접속된다. 데이터 네트워크 허브(14)는 엘리먼트 제어 유닛(12)으로부터 수신된 데이터에 지속적으로 로깅한다.

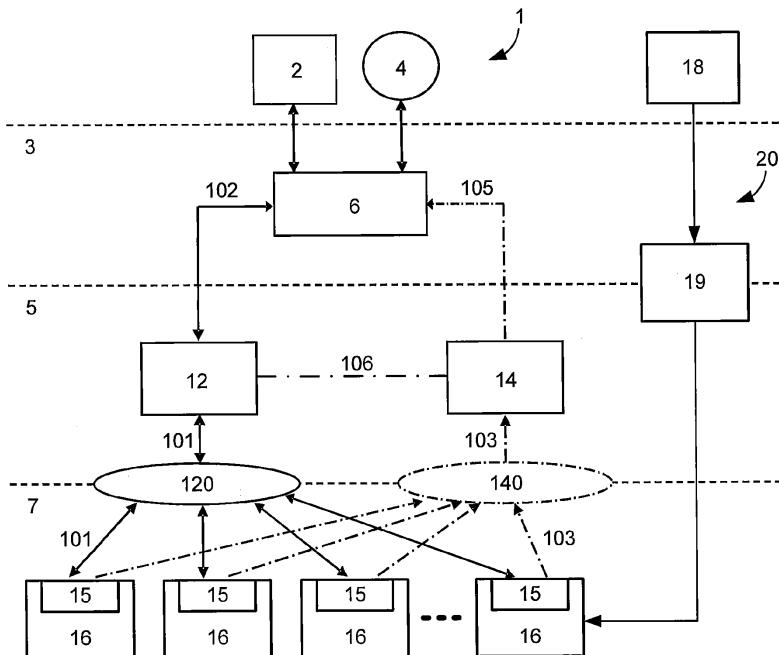
[0107] 데이터 네트워크 허브(14)는, 긴 동작 시간 기간, 예를 들어 대략 수개월 또는 수년에 걸친 모든 서브시스템들(16)로부터의 매우 큰 양들의 저레벨 데이터의 저장을 위해 충분한, 매우 큰 데이터 저장 용량을 포함한다. 데이터 네트워크 허브(14)에 의해 저장된 데이터는, 바람직하게는 타임 스탬프 및 PJ 식별자를 이용하여, 조직 및 태깅된다. 저장된 데이터는 데이터 네트워크 허브(14)로부터 검색되고, 그리고 바람직하게는 오프-라인으로 분석 및 필터링될 수 있다.

[0108] 제어 네트워크(120)를 통한 제어 통신들로부터의, 데이터 네트워크(140)를 통한 데이터 수집의 분리는, 제어 네트워크에 걸쳐 통신들을 절충함이 없이, 데이터 네트워크에 걸친 고볼륨 데이터의 고빈도 수집을 가능하게 한다. 제어 네트워크는, 제어 네트워크(120) 상에서의 혼잡을 제어함으로써, 그리고 데이터 네트워크(140) 상에 존재하는 고볼륨 트래픽을 회피함으로써, 준 실시간으로 동작하는 것이 가능하게 된다. 엘리먼트 제어 유닛(12) 내 PJ 실행 및 데이터 네트워크 허브(14) 내 데이터 관리 및 저장 기능들의 분리는, 엘리먼트 제어 유닛(12)에 의한 PJ들의 프로세싱을 절충함이 없이 데이터 네트워크 허브(14)에 의한 데이터의 고볼륨들의 프로세싱을 가능하게 한다. 제어 및 데이터 수집 및 관리의 분리는, 2개의 시스템들 사이의 상호작용을 고려할 필요 없이 더 단순한 설계를 가능하게 한다.

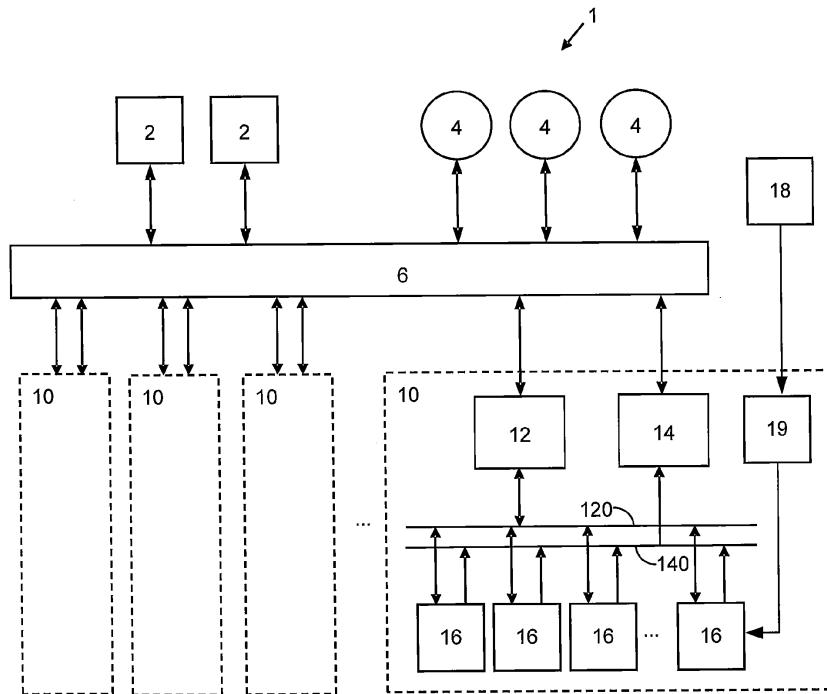
[0109] 본 발명은 상기 논의된 특정 실시예들을 참조하여 기술되었다. 이들 실시예들은, 본 발명의 범주 및 사상으로부터 벗어남이 없이 당업자들에게 잘 알려진 다양한 수정들 및 대안적인 형태들을 허용하는 것으로 인식될 것이다. 따라서, 특정 실시예들이 기술되지만, 이들은 단지 예들이며, 첨부된 청구범위에서 규정되는 본 발명의 범주에 대해 제한적이지 않다.

도면

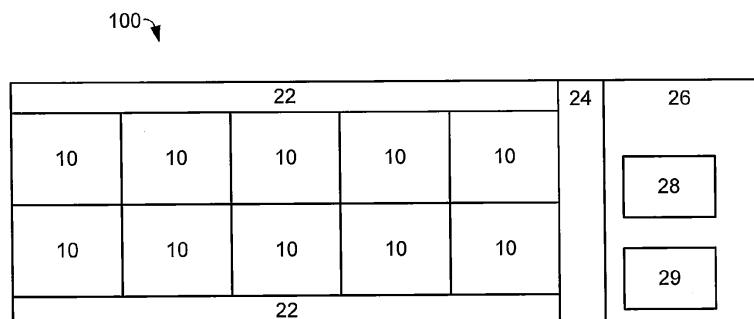
도면1



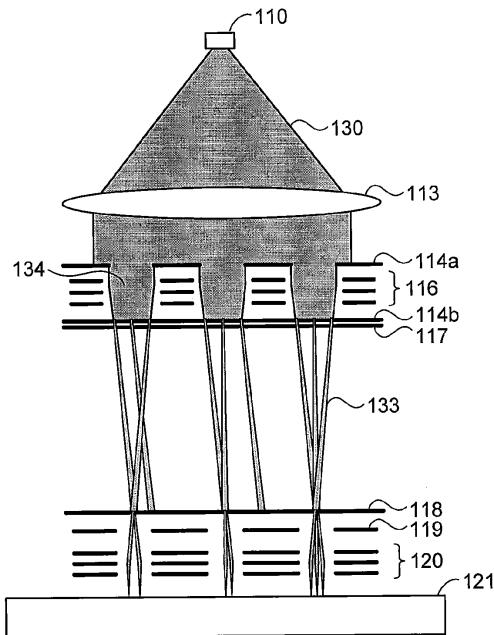
도면2



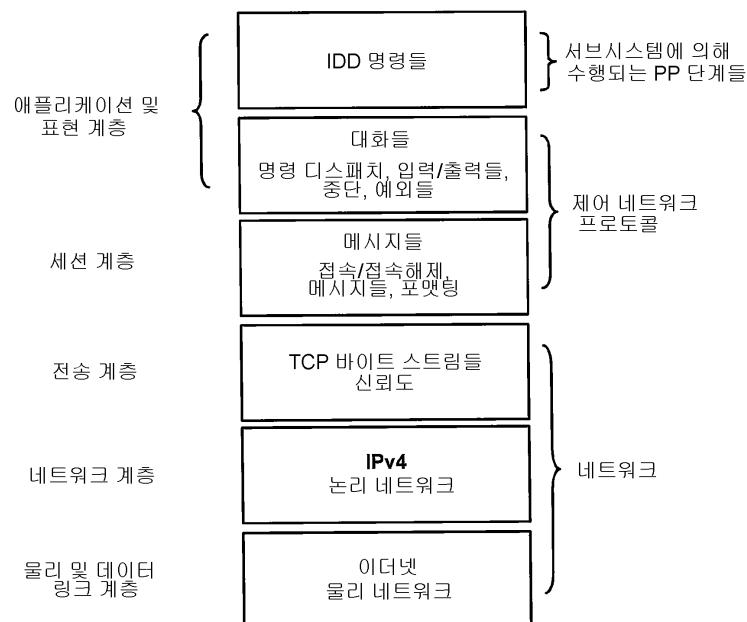
도면3



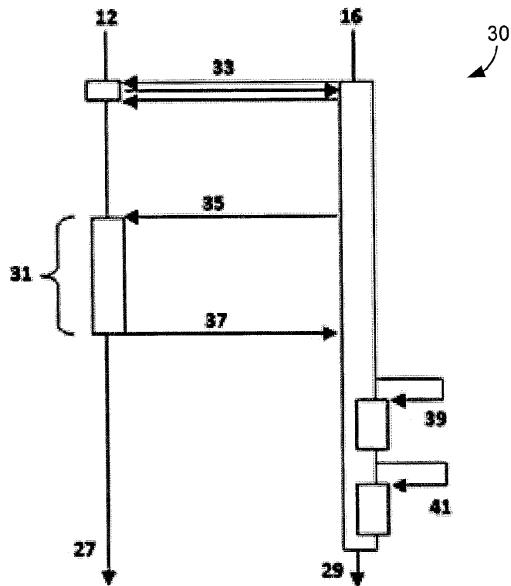
도면4



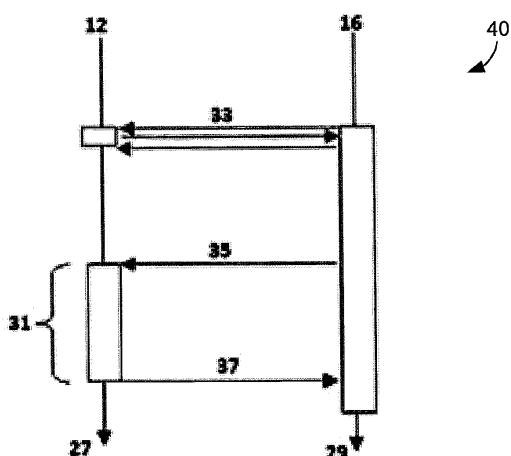
도면5



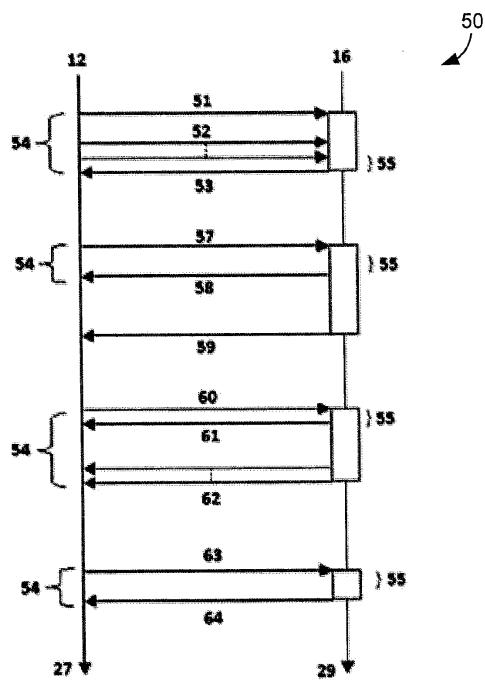
도면6a



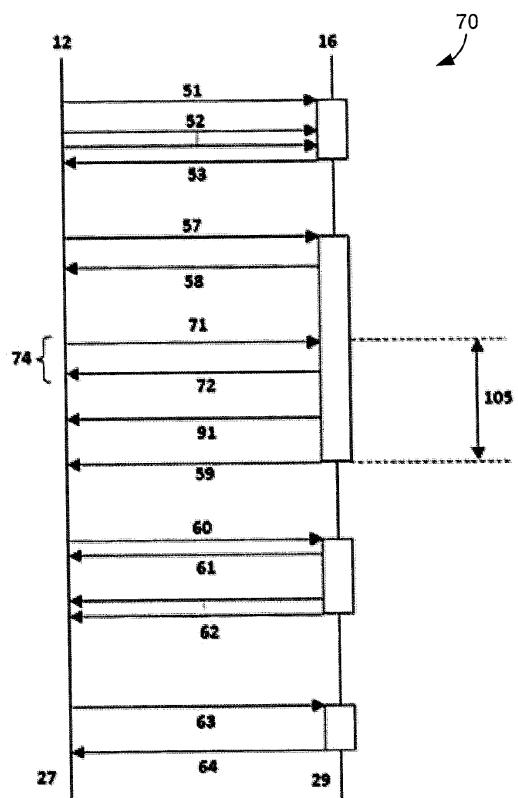
도면6b



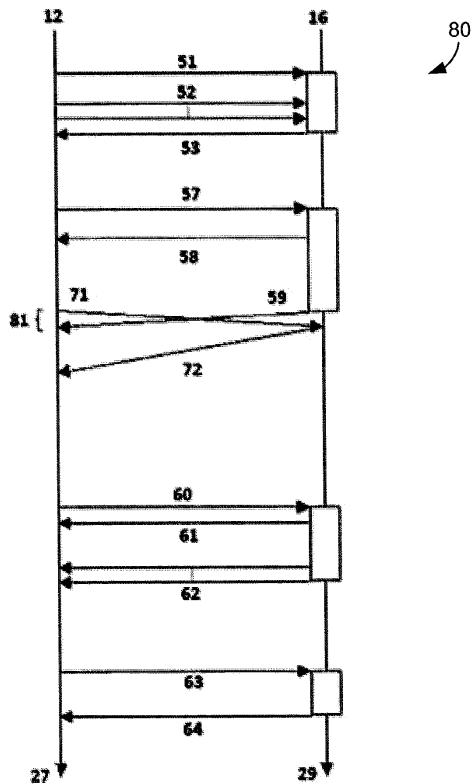
도면6c



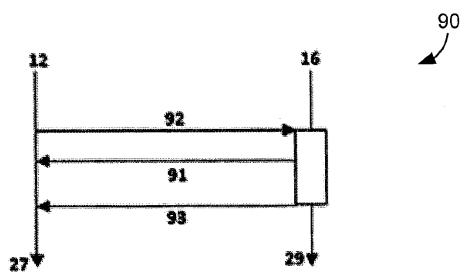
도면6d



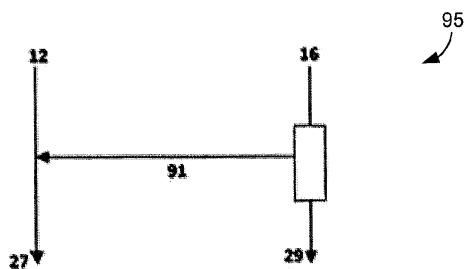
도면6e



도면6f



도면6g



도면6h

