(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) 。Int. Cl. 6
H04L 29/06

(45) 공고일자 2001년10월26일
(11) 등록번호 10 - 0297219
(24) 등록일자 2001년05월18일

(21) 출원번호 10 - 1998 - 0037116
(22) 출원일자 1998년09월09일

(65) 공개번호 특1999 - 0029655
(43) 공개일자 1999년04월26일

(30) 우선권주장 97 - 244430 1997년09월09일 일본(JP)

(73) 특허권자
캐논 가부시끼가이샤
미다라이 후지오
일본 도꾜도 오오따꾸 시모마루꼬 3쪼메 30방 2고

(72) 발명자
이소다 다까시
일본 도꾜도 오오따꾸 시모마루꼬 1쪼메 12 - 20캐논 보누루 시모마루꼬내
시무라 아끼히로
일본 가나가와껭 가와사끼시 나까하라꾸 가미꼬다나까 3쪼메 16 - 14 - 10

(74) 대리인
구영창
장수길
주성민

심사관 : 여인홍

(54) 통신 제어 방법 및 장치, 및 통신 시스템

요약

순차(queued) 실행 명령을 큐잉하기 위한 큐, 기록 명령을 실행하기 위한 즉시 실행 에이전트, 및 판독 명령을 실행하기 위한 순차 실행 에이전트를 갖는 프린터를 제공한다. 상기 즉시 실행 에이전트는 수신된 기록 명령을 즉시 실행하고, 호스트 내에 데이타를 기록한다. 순차 실행 에이전트는 큐로부터의 판독 명령을 픽업하여 호스트로부터 데이타를 판독한다. 호스트는 프린터로부터의 데이터 전송 요청을 큐에 부가하여, 데이타 전송 요청에 기초하여 프린터에 기록 명령을 발행하고, 애플리케이션으로부터의 인쇄 데이타 전송 요청 등을 기초로 하여 프린터로 판독 명령을 발행한다. 독립 전이중 채널(full - duplex channels)이 양방향으로 제공될 수 있다. 또한, 기록 명령이 즉시 처리될 수 있다.

대표도

도 1

명세서

도면의 간단한 설명

도 1은 타겟(프린터)의 블럭도.

도 2는 이니시에이터(호스트 컴퓨터)의 블럭도.

도 3a 및 3b는 ORB의 일반적인 형식을 도시하는 도면.

도 4는 QUEUE DEPTH 명령 ORB의 형식을 도시하는 도면.

도 5는 DATA TRANSFER 명령 ORB의 형식을 도시하는 도면.

도 6는 REQUESTED READ 명령 ORB의 형식을 도시하는 도면.

도 7은 DIRECT STATUS RESPONSE 명령 ORB의 형식을 도시하는 도면.

도 8a 및 8b는 ACQUIRE DEVICE RESOURCE 명령 ORB의 형식을 도시하는 도면.

도 9는 데이타 리소스 해제 명령 ORB의 형식을 도시하는 도면.

도 10은 BASIC DEVICE STATUS 명령 ORB의 형식을 도시하는 도면.

도 11a 및 11b는 상태 블럭의 일반적인 형식을 도시하는 도면.

도 12는 QUEUE DEPTH 상태 블럭의 형식을 도시하는 도면.

도 13은 DATA TRANSFER 상태 블럭의 형식을 도시하는 도면.

도 14는 발생된 데이타 전송 요청에 응답하여 이니시에이터에 의해 실행된 처리 절차를 도시하는 흐름도.

도 15는 DOORBELL 레지스터에서의 기록시 타겟의 페치 에이전트에 의해 실행된 처리 절차를 도시하는 도면.

도 16은 페치 에이전트로부터 ORB 수신시 실행 에이전트에 의해 실행된 처리 절차를 도시하는 흐름도.

도 17은 발생된 데이타 전송 요청에 응답하여 타겟에 의해 실행된 처리 절차를 도시하는 흐름도.

도 18은 상태 레지스터에서의 기록시 이니시에이터에 의해 실행된 처리 절차를 도시하는 흐름도.

도 19는 이니시에이터(호스트 컴퓨터)로부터 타겟(프린터)으로의 데이타 전송 시퀀스를 도시하는 도면.

도 20은 READ REQUEST 상태를 사용하여 타겟으로부터 이니시에이터로의 데이타 전송시 시퀀스를 도시하는 도면.

도 21은 DIRECT 상태를 사용하여 타겟으로부터 이니시에이터로 데이타 전송시 시퀀스를 도시하는 도면.

도 22는 다중 채널을 제공하는 타겟의 블럭도.

도 23은 다중 채널을 제공하는 이니시에이터의 블럭도.

도 24는 다중 채널의 DATA TRANSFER 명령 ORB 형식을 도시하는 도면.

도 25는 다중 채널의 REQUESTED READ 명령 ORB 형식을 도시하는 도면.

도 26은 다중 채널의 DIRECT STATUS RESPONSE 명령 ORB의 형식을 도시하는 도면.

도 27a 및 27b는 다중 채널의 ACQUIRE DEVICE RESOURCE 명령 ORB 형식을 도시하는 도면.

도 28은 다중 채널의 RELEASE DEVICE RESOURCE 명령 ORB의 형식을 도시하는 도면.

도 29는 다중 채널의 ABDICATE DEVICE RESOURCE 명령 ORB의 형식을 도시하는 도면.

도 30은 BASIC DEVICE STATUS 명령 ORB의 형식을 도시하는 도면.

도 31a 및 31b는 OPEN CHANNEL REQUEST 응답의 형식을 도시하는 도면.

도 32a 및 32b는 CLOSE CHANNEL REQUEST 응답의 형식을 도시하는 도면.

도 33a 및 33c는 다중 채널의 상태 블럭의 일반적인 형식을 도시하는 도면.

도 34는 DATA TRANSFER 상태 블럭의 형식을 도시하는 도면.

도 35는 다중 채널의 DIRECT 상태 블럭의 형식을 도시하는 도면.

도 36은 다중 채널의 DEVICE RESOURCE ACQUIRE 상태 블럭의 형식을 도시하는 도면.

도 37은 다중 채널의 ABDICATE DEVICE RESOURCE 상태 블럭의 형식을 도시하는 도면.

도 38은 다중 채널의 BASIC DEVICE 상태 블럭의 형식을 도시하는 도면.

도 39a는 상태 레지스터에 기록시 다중 채널의 이니시에이터에 의해 실행된 처리 절차를 도시하는 흐름도.

도 39b는 DOORBELL 레지스터에 기록시 다중 채널의 타겟에 의해 실행된 처리 절차를 도시하는 흐름도.

도 40은 IEEE1394 인터페이스를 사용하는 인쇄 시스템의 하드웨어 구성을 도시하는 블럭도.

<도면의 주요 부분에 대한 부호의 설명>

100: 프린터

101: 1394 인터페이스

104: 프리페치 큐

105: 순차 실행 에이전트

106: 즉시 실행 에이전트

203: HPT 프로세서

209: 애플리케이션

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 호스트 컴퓨터 및 프린터 등의 장치들을 접속하기 위한 통신 제어 방법 및 장치에 관한 것이다.

최근, 컴퓨터와 주변 장치 또는 주변 장치들를 접속하기 위해 IEEE1394 인터페이스가 사용된다. IEEE1394 인터페이스는 중앙 인터페이스(Centronics interface) 등의 핸드-세이크 스킴(hand-shake scheme)과 비교하여 고속이고 양방향 통신이 가능하다. 또한, IEEE1394 인터페이스는 메모리 버스 모델 인터페이스로, IEEE1394 인터페이스를 통하여 접속된 장비는 접속된 장비에 대하여 지정된 어드레스로부터 데이타를 판독할 수 있고 상기의 지정된 어드레스에서 데이타를 기록할 수 있다.

IEEE1394는 많은 응용 분야에 사용되고 있는 물리층 및 링크층의 프로토콜 을 한정하나, 장비 마다 프로토콜을 세부적으로 한정하는 것은 아니다. 이러한 이유로, 물리층 및 링크층으로서 IEEE1394를 사용하는 SBP (Serial Bus Protocal)-2 등의 트랜스포트층의 프로토콜이 제안되어 왔다. 트랜스토트층은 데이타 전송 기능을 애플리케이션에 제공하고, 이 층을 사용하는 애플리케이션은 상호 데이타를 교환할 수 있다.

이 프로토콜 SBP-2는 IEEE1394의 메모리 버스 모델 특성을 이용하고, 이 프로토콜로 인해, 데이타 수신측은 자신의 리소스가 이용가능해질때 데이타를 수신할 수 있다.

SBP-2에서, 데이타가 전송될때 전송측은 소위 로그인(login) 동작을 수행하여 통신 상대와의 사이에 채널을 설치하게 된다. 이 경우에서, 로그인 측은 이니시에이터(initiator)라 하고, 그 상대는 타겟(target)이라 한다. 데이타 전송은 이니시에이터로부터의 지시(instruction)에 따라 이니시에이터로/부터 데이타를 판독/기록하는 방법으로 수행된다. 이 방식에서, 이니시에이터는 로그인 후에 전송될 데이타의 기억 어드레스, 크기 등을 포함하는 ORB(Operation Request Block)를 생성하고 ORB의 어드레스를 타겟에 통지한다. 타겟은 ORB 내에 기록된 어드레스 및 크기를 기준으로 하여 자신의 리소스가 이용가능하게 될때 이니시에이터로부터 데이타를 판독하여 판독된 데이타를 처리하거나 또는 데이타를 기록한다. 이러한 처리후에, 타겟은 이니시에이터에게 처리 상태를 통지하는 상태 블럭(status block)을 발생한다.

SBP-2를 사용하는 통신이 IEEE1394를 기초로 이루어진 경우, 특히 SBP-2가 이니시에이터로서 역할하는 호스트 컴퓨터 등의 데이타 소스로부터 타겟으로서 역할하는 프린터 장치 등의 주변 장치로의 데이타 전송에 이용될 경우 두가지 문제점이 제기된다.

(1) 전이중(full-dulpex) 통신에 기인하여 절차가 복잡해진다.

SBP-2에서, 데이타 전송은 기본적으로 이니시에이터에 의해 관리되고, 타겟은 이니시에이터에 대해 데이타를 비동기적으로 전송할 수 없다. SBP에서, 타겟이 이니시에이터쪽으로 데이타를 전송하기를 원할 경우, 언솔리사이티드 상태(unsolicited status)를 이용하여 데이타 판독 요청(request)을 이니시에이터에 송신한다. 이니시에이터는 상기 요청에 응답하여 현안의(pending) ORB(이니시에이터로부터 타겟 등으로의 데이타 전송 요청을 포함한) 리스트의 마지막에 상기 생성된 ORB를 추가한다. 이들 ORB는 가장 오래된 것으로부터 차례로 처리된다. 이러한 이유로, 이니시에이터측의 ORB 처리만이 진행되고, 이니시에이터가 타겟으로부터 데이타 판독 요청에 응답하여 생성된 ORB를 처리하는 경우에만, 타겟이 전송 데이타를 이니시에이터에 전송할 수 있다. 즉, 타겟으로부터 이니시에이터로의 데이타 전송 타이

밍은 타겟으로부터 이니시에이터로의 판독 요청 발행 타이밍이 아니라 현안의 ORB를 처리하기 의해 필요한 시간 만큼의 타이밍으로 지연된다. 그 결과, 양방향, 비동기식 데이타 전송이 이루어질 수 없다. 타겟으로부터 이니시에이터로 전송될 데이타가 비동기식으로 생성된 경우 예를 들어, 타겟이 프린터이고 그 프린터에서 에러가 생성한 경우, 이니시에이터에게로 즉시 전송되어야 할 데이타는 실시간(real time) 내에 전송될 수 없다.

이러한 이유로, 프린터에 의해 비동기적으로 생성된 데이타를 실시간 내에 호스트로 전송하기 위해서, 프린터는 이니시에이터로서 로그인 절차를 수행해야 하고, 호스트 컴퓨터를 타겟으로 하여 데이타 전송을 수행해야 한다.

이러한 방법으로, 호스트 컴퓨터 및 프린터가 상호 로그인인 경우, 이들 둘다 이니시에이터 및 타겟으로 역할하며, 이들은 이니시에이터 및 타겟 둘다로서 자신의 프로세스를 처리해야만 한다. 또한 프린터는 로그인을 수행해야 한다.

화상을 처리하는 프린터 등의 주변 장치는 화상 처리를 위한 대용량의 메모리 리소스 및 프로세서 리소스를 소비한다. 이러한 이유로, 장치 구성을 간단히 함으로써 비용을 줄이고 빠르게 처리하기 위해, 화상 처리와는 다른 목적으로 사용된 리소스를 가능한한 줄여야만 한다. 그러나, 프린터가 상술한 바와 같이 많은 프로세스를 처리해야 하는 경우, 많은 리소스들이 소모됨으로써 비용 감소 및 효율적인 처리를 방해하게 된다.

한편, 호스트 컴퓨터와 프린터 사이의 데이타 흐름은 인쇄 데이타 및 그 처리 상태와 같이 상호 연관되어 있다. 만약 채널이 독립적 로그인 프로세스에 의해 양방향으로 설정된다면, 그 데이타 및 응답들이 상호 연관되어야 하고, 이들에 대한 새로운 처리 프로토콜이 부가되어야만 한다.

이러한 방법으로, IEEE1394 및 SBP - 2를 호스트 컴퓨터 및 프린터 간의 통신에 직접 적용하는 것은 적합하지 않고, 개별 장치들 내에 요구된 리소스를 줄이기 어려워 효율을 향상시키기는 어렵다.

(2) 다중 채널(multi - channel)을 실현할 수 없다.

최근, 다양한 기능을 결합시킨 다기능의 머쉰이 주변 장치로서 널리 사용된다. 예를 들어, 호스트 컴퓨터를 스캐너, 프린터, 및 팩스로서 사용할 수 있게 하는 디지탈 다기능 머쉰이 공지되어 있다. 이러한 장치가 사용된 경우, 복수의 독립 채널을 통하여 통신이 기능 단위로 이루어진다면 복수개의 기능들이 동시에 사용될 수 있다.

그러나, SBP - 2는 다중 채널을 제공할 수 없기 때문에, 이러한 단위 기능을 동시에 사용하기는 어렵다.

SBP - 2와는 다른 임의 프로토콜이 비동기적으로 생성된 데이타를 전송할 수 있고 다중 채널을 실현할 수 있다. 그러나, 이러한 프로토콜은 IEEE1394의 특성을 메모리 버스 모델로서 사용할 수 없다. 즉, 이러한 프로토콜이 호스트 및 프린터 간의 통신에 적용된 경우, 프린터의 편의대로 데이타 전송을 수행할 수 없고, 호스트 컴퓨터는 프린터 상태를 감시하면서 데이타 전송을 수행해야만 한다.

발명이 이루고자 하는 기술적 과제

본 발명은 상술한 종래 기술을 고려하여 이루어진 것으로서, 싱글 로그인(single login) 처리에 의한 전이중 통신(비동기식 양방향 통신)을 수행할 수 있고, 데이타 교환에 필요한 프로세스, 메모리 등의 리소스를 효율적으로 사용할 수 있는 통신 제어 방법 및 장치, 및 상기 방법을 사용하는 프린터 장치를 제공하는 데 그 목적이 있다.

본 발명의 다른 목적은 리소스가 이용가능하게 되는 대로 즉시 한 이니시에이터에 의해 준비된 데이타를 타겟으로 하여금 판독할 수 있게 하고, 타겟의 편의대로 데이타 전송에 의해 이니시에이터가 점유되는 것을 방지하는 통신 제어 방법 및 장치, 및 상기 방법을 사용한 프린터 장치를 제공하는 데 있다.

본 발명의 또 다른 목적은 다중 채널을 실현할 수 있는 통신 제어 방법 및 장치, 및 상기 다중 채널을 사용하는 프린터 장치를 제공하는 데 있다.

상기 목적을 달성하기 위해 본 발명은 다음의 구성을 포함한다.

즉, 이니시에이터의 기억 영역을 액세스시 타겟으로부터 데이타를 교환하는 통신 제어 방법에 있어서,

상기 이니시에이터는 기억 영역에 대한 판독 및 기록 액세스에 대응하는 명령을 상기 타겟에 의해 보유될 수 있는 판독 및 기록 명령의 수를 초과하지 않도록 상기 타겟으로 송신하며,

상기 타겟은 상기 수신된 판독 및 기록 명령을 상이한 큐 내에 보유하고 상기 보유된 명령을 독립적으로 프로세스하는 것을 특징으로 하는 통신 제어 방법을 제공한다.

또한, 본 발명은 이니시에이터의 기억 영역 액세스시 타겟으로부터 데이타를 교환(exchanging)하는 통신 제어 방법에 있어서,

상기 타겟은 전송될 데이타의 크기가 선정된 크기를 초과하는지 체크하여, 전송될 데이타의 크기가 상기 선정된 크기를 초과하는 경우 상기 기억 영역 내에 기록 명령을 발행하도록 상기 이니시에이터에 대해 요청하며, 전송될 상기 데이타의 크기가 상기 선정된 크기를 초과하지 않는 경우 상기 데이타를 상기 이니시에이터로 송신하며,

상기 이니시에이터는 상기 타겟으로부터 기록 명령 발행 요청 수신시 기록 명령을 발행하는 것을 특징으로 하는 통신 제어 방법을 제공한다.

또한, 본 발명은 이니시에이터의 기억 영역 액세스시 타겟으로부터 데이타를 교환하기 위한 통신 시스템에 있어서,

상기 이니시에이터는 상기 기억 영역에 대한 판독 및 기록 액세스에 대응하는 명령을 상기 타겟에 의해 보유될 수 있는 판독 및 기록 명령의 수를 초과하지 않도록 상기 타겟으로 전송하고,

상기 타겟은 수신된 판독 및 기록 명령을 상이한 큐에 보유하고, 상기 보유된 명령을 독립적으로 프로세스하는 것을 특징으로 하는 통신 시스템을 제공한다.

또한, 본 발명은 이니시에이터의 기억 영역 액세스시 타겟으로부터 데이타를 교환하기 위한 통신 시스템에 있어서,

상기 타겟이 전송될 데이타의 크기가 선정된 크기를 초과하는지 여부를 체크하여, 상기 전송될 데이타의 크기가 상기 선정된 크기를 초과하는 경우 상기 이니시에이터가 상기 기억 영역 내에 기록 명령을 발행하도록 요청하고, 전송될 상기 데이타의 크기가 선정된 크기를 초과하지 않는 경우 상기 데이타를 상기 이니시에이터에 송신하며,

상기 이니시에이터는 타겟으로부터 기록 명령 발행 요청 신호 수신시 기록 명령을 발행하는 것을 특징으로 하는 통신 시스템을 제공한다.

또한 본 발명은, 통신에 의해 접속된 상기 타겟으로부터 메모리 내의 기억 영역을 액세스시 타겟과 데이타를 교환하는 통신 제어 방법에 있어서,

상기 타겟으로부터의 자발적인 요청(spontaneous request)을 수신하여 큐 내에 상기 요청을 큐잉(queing)하는 큐잉 단계, 및

사기 타겟에 의해 보유될 수 있는 판독 및 기록 명령의 수를 초과하지 않도록 애플리케이션 또는 상기 타겟으로부터의 요청에 응답하여 상기 기억 영역에 판독 및 기록 명령을 생성하여 전송하는 명령 생성 단계를 포함하는 것을 특징으로 하는 통신 제어 방법을 제공한다.

또한, 본 발명은 통신을 통하여 접속된 이니시에이터의 기억 영역 액세스시 상기 이니시에이터와 데이타를 교환하는 통신 제어 방법에 있어서,

선정된 크기를 갖는 큐 내에서 상기 이니시에이터로부터 수신된 판독 명령을 큐잉하는 큐잉 단계,

상기 큐로부터의 판독 명령을 픽업(picking up) 하고 수행하는 순차 실행(queued execution) 단계,

상기 이니시에이터로부터 수신된 기록 명령을 수신 후 즉시 수행하는 즉시 실행 단계, 및

데이타 전송 요청을 상기 이니시에이터에 발행하는 전송 요청 단계를 포함하는 것을 특징으로 하는 통신 제어 방법을 제공한다.

또한, 본 발명은 기억 영역을 통하여 타겟과 데이타를 교환하기 위한 통신 제어 장치에 있어서,

타겟과 통신하기 위한 수단,

상기 기억 영역을 포함하는 메모리,

상기 타겟으로부터 자발적인 요청을 큐잉하기 위한 큐 관리 수단, 및

애플리케이션 또는 상기 타겟으로부터의 요청에 응답하여 상기 타겟에 의해 보유될 수 있는 판독 및 기록 명령의 수를 초과하지 않도록 상기 기억 영역에 관련된 판독 및 기록 명령을 생성 및 전송하는 명령 생성 수단을 포함하는 것을 특징으로 하는 통신 제어 장치를 제공한다.

또한, 본 발명은 이니시에이터의 기억 영역을 액세스함으로써 상기 이니시에이터와 데이타를 교환하기 위한 통신 제어 장치에 있어서,

상기 이니시에이터와 통신하기 위한 수단,

상기 이니시에이터로부터 수신된 판독 명령을 보유하는 선정된 크기의 큐,

상기 이니시에이터로부터 판독 명령을 픽업하고 수행하는 순차 실행 수단,

상기 이니시에이터로부터 수신된 기록 명령을 수신 후 즉시 수행하기 위한 즉시 실행 수단, 및

데이타 전송 요청을 상기 이니시에이터에 발행하기 위한 전송 요청 수단을 포함하는 것을 특징으로 하는 통신 제어 장치를 제공한다.

또한, 본 발명은 통신에 의해 타겟으로부터 기억 영역을 액세스함으로써 타겟으로부터 데이타를 교환하는 통신 제어 프로그램을 기억하는 컴퓨터 판독가능 기억 매체에 있어서, 상기 프로그램이 상기 타겟으로부터 자발적인 요청을 큐잉하기 위한 큐 관리 수단, 및

애플리케이션 또는 상기 타겟으로부터의 요청에 응답하여 상기 타겟에 의해 보유될 수 있는 판독 및 기록 명령의 수를 초과하지 않도록 상기 기억 영역에 관련된 판독 및 기록 명령을 생성 및 전송하는 명령 생성 수단을 포함하는 것을 특징으로 하는 컴퓨터 판독가능 기억 매체를 제공한다.

또한, 본 발명은 통신에 의해 접속된 이니시에이터의 기억 영역을 액세스함으로써 데이타를 교환하기 위한 통신 제어 프로그램을 기억하는 컴퓨터 판독가능 기억 매체에 있어서,

상기 이니시에이터로부터 수신된 판독 명령을 선정된 용량을 갖는 큐내에서 큐잉하기 위한 큐 관리 수단,

상기 큐로부터 상기 판독 명령을 픽업 및 실행하는 순차 실행 수단,

상기 이니시에이터로부터 수신된 기록 명령을 수신 후 즉시 실행하기 위한 즉시 실행 수단, 및

데이타 전송 요청을 상기 이니시에이터에 발행하기 위한 전송 요청 수단을 포함하는 것을 특징으로 하는 컴퓨터 판독가능 기억 매체를 제공한다.

또한, 본 발명은 타겟에 의해 보유될 수 있는 판독 및 기입 명령의 수를 초과하지 않도록 이니시에이터가 기억 영역에 대한 판독 및 기록 액세스에 대응하는 명령을 타겟에 전송하고, 상기 타겟은 수신된 판독 및 기록 명령을 상이한 큐 내에 보유하고, 보유된 명령을 독립적으로 처리하는 통신 제어 방법을 사용하는 인쇄 시스템에 있어서,

이니시에이터로서 동작하는 호스트 장치가 타겟으로서 역할하는 프린터 장치에 접속되고, 상기 프린터 장치는 상기 호스트 장치로부터 인쇄 데이타를 수신하고 상기 수신된 인쇄 데이타를 기준하여 출력하며, 상기 호스트 장치는 상기 프린터 장치의 상태 정보를 수신하는 것을 특징으로 하는 인쇄 시스템을 제공한다.

또한 본 발명은 타겟으로부터 자발적인 요청을 수신하여 상기 요청을 큐에서 큐잉하는 큐잉 단계, 및 타겟에 의해 보유될 수 있는 판독 및 기록의 수를 초과하지 않도록 애플리케이션 또는 타겟으로부터 요청에 응답하여 상기 기억 영역으로 판독 및 기록 명령을 생성하여 전송하는 명령 생성 단계를 포함하는 통신 제어 방법으로 인쇄 데이타를 타겟에 전송하고 타겟으로부터 상태 정보를 수신하는 송신하고, 상기 타겟으로부터 상태 정보를 수신하기 위한 인쇄 제어 장치를 제공한다.

또한 본 발명은 선정된 크기를 갖는 큐에서 이니시에이터로부터 수신된 판독 명령을 큐잉하는 큐잉 단계, 큐로부터 판독 명령을 픽업하여 실행하는 순차 실행 단계, 이니시에이터로부터 수신된 기록 명령을 수신후 즉시 실행하는 즉시 실행 단계, 및 상기 이니시에이터에 데이타 전송 요청을 발행하는 전송 요청 단계를 포함하는 통신 제어 방법에 의해 이니시에이터로부터 인쇄 데이타를 수신하고, 상기 이니시에이터로 상태 정보를 전송하기 위한 인쇄 장치를 제공한다.

본 발명의 다른 특성 및 장점들은 첨부된 도면을 참조로 한 다음의 설명으로부터 더욱 분명해질 것이며, 도면에서 동일하거나 유사한 부분에는 동일한 참조 부호를 부가한다.

발명의 구성 및 작용

[제1 실시예]

IEEE1394를 통해 호스트 컴퓨터와 프린터를 접속하는 프린터 시스템이 본 발명의 제1 실시예로서 설명된다. 이 시스템에서 데이타 전송은 IEEE1394에 의거한 SBP-2를 이용하는 본 발명의 프로토콜(이후, HPT라 함)에 따라 행해진다. 도 40은 프린터 시스템 내의 하드웨어 구성을 나타내고 있다.

<시스템의 하드웨어 구성>

도 40에서 호스트 컴퓨터(200)는 도형(figures), 이미지, 문자, 표(표 계산 등을 포함함) 등, ROM(3)의 프로그램 ROM 영역에 저장된 문서 처리 프로그램을 기초로 포함하고 있는 문서를 처리하는 CPU(1)를 구비하고 있다. CPU(1)는 시스템 버스(4)에 접속된 각각의 디바이스를 총괄적으로 제어한다. ROM(3)의 프로그램 ROM 영역은 CPU(1)의 제어 프로그램을 저장하고 있고, ROM(3)의 폰트 ROM 영역은 문서 처리에 사용되는 폰트 데이터 등을 저장하고 있으며, ROM(3)의 데이터 ROM 영역은 문서 처리 등의 실행시 사용되는 여러 데이터를 저장하고 있다. RAM(2)은 작업 영역으로서 CPU 등의 주 메모리의 역할을 한다. 프로그램은 RAM(2)에 저장될 수도 있다. RAM(2)에는 전송 데이터 버퍼 및 ORB들을 저장하기 위한 시스템 메모리가 확보된다.

키보드 콘트롤러(KBC;5)는 키보드(9) 및 포인팅 시스템(도시 생략)으로부터의 키입력을 제어한다. CRT 콘트롤러(CRTC;6)는 CRT 디스플레이(CRT;10) 상의 디스플레이를 제어한다. 메모리 콘트롤러(MC;7)는 부트 프로그램, 여러 애플리케이션, 폰트 데이터, 유저 파일, 에디트 파일 등을 저장하는 하드디스크(HD), 플로피디스크(FD) 등의 외부 메모리(11)와의 액세스를 제어한다. 1394 인터페이스(8)는 IEEE1394 표준에 따른 프린터(100)에 접속되어 프린터(100)와의 통신 제어를 구현한다. 예를 들어 CPU(1)는 출력 라인 폰트 데이터를 RAM(2)에 할당된 디스플레이 정보 RAM 영역에 맵핑하여(rasterizes) CRT(10)상에서의 WYSIWYG 환경을 실현한다. CPU(1)는 마우스 커서(도시생략) 등에 의해 지정된 명령에 따른 등록된 여러 윈도우를 열어 각종 데이터 처리를 실행한다.

프린터(100)에서 프린터 CPU(12)는 ROM(13)의 프로그램 ROM 영역에 저장된 제어 프로그램 또는 외부 메모리(14)에 저장된 제어 프로그램에 기초하여 시스템 버스(15)에 접속된 각종 디바이스와의 액세스를 총괄적으로 제어하고, 인쇄부 인터페이스(16)를 통해 접속되는 인쇄부(프린터 엔진)에 출력 정보로서 화상 신호를 출력한다. ROM(13)의 프로그램 ROM 영역에는 여러 에이전트(후술됨)를 구현하는 CPU(12)를 위한 제어프로그램이 저장된다. ROM(13)의 폰트용 ROM 영역에는 출력 정보의 생성시 사용되는 폰트 데이터 등이 저장되고, ROM(13)의 데이터용 ROM 영역에는 하드디스크 등의 외부 메모리(14)를 전혀 갖지 않는 프린터의 경우에 호스트 컴퓨터 상에서 사용되는 정보 등이 저장된다. CPU(12)는 1394 인터페이스(18)를 통해 호스트 컴퓨터와 통신할 수 있고, 프린터의 정보 등을 호스트 컴퓨터(200)에 통지할 수 있다.

RAM(19)은 CPU(12) 등의 주 메모리로서 작업영역의 역할을 하며, 그 메모리 용량은 확장 포트(도시 생략)에 접속된 옵션 RAM을 이용하여 확장될 수 있다. RAM(19)은 출력 정보 맵핑 영역(output information mapping area), 환경 데이터 저장 영역, NVRAM 등으로 사용된다.

메모리 콘트롤러(MC)(20)는 하드디스크(HD), IC 카드 등의 상기한 외부 메모리(14)와의 액세스를 제어한다. 외부 메모리(14)는 옵션으로서 접속되어 폰트 데이터, 에뮬레이션 프로그램, 폼 데이터(form data) 등을 저장한다. 조작 패널(콘솔;1002)에는 동작 스위치, LED 지시계 등이 제공된다. 외부 메모리의 수는 하나로 제한되는 것은 아니고, 프린터는 내부 폰트 뿐만 아니라 옵션 폰트를 포함하는 옵션 폰트 카드를 포함하는 다수의 메모리가 접속될 수도 있도록 하나 이상의 외부 메모리를 포함할 수도 있다. 더욱이, NVRAM(도시생략)이 부가될 수도 있으며 조작 패널(1002)로부터의 프린터 모드 셋업 정보를 저장할 수도 있다.

<이니시에이터(Initiator)의 구성>

도 1 및 도 2는 프린터를 타겟으로 하고 상기한 하드웨어 구성의 개시로서 호스트 컴퓨터(200)를 이니시에이터로 사용하는 통신 시스템을 나타내고 있다. 본 실시예에서 이러한 구성은 호스트 컴퓨터와 프린터에서의 CPU에 의해 프로그램이 실행됨으로써 구현된다. 도 2에 도시된 이니시에이터를 먼저 설명한다.

도 2에 있어서 이니시에이터의 기능을 하는 호스트 컴퓨터에서는, 애플리케이션인 프린터 드라이버(209)는 HPT 프로세서(203)를 통해 프린터에 대한 데이터 전송 요구를 발행하고 그 프린터로부터 응답(reply)을 수신한다.

HPT 프로세서(203)는 ORB 관리부(206)를 포함하고 있다. ORB 관리부(206)는 시스템 메모리(208)에서 생성된 ORB들을 관리한다. ORB는 호스트 컴퓨터로부터 프린터로 또는 프린터로부터 컴퓨터로 전송되는 데이터 버퍼 의 어드레스나, 사이즈 등을 저장하는 블록이다. ORB들은 순차로 링크되어 ORB 리스트를 형성한다. 이러한 ORB들의 경우, 다음의 처리 원칙이 있다:

(1) ORB 리스트의 ORB들은 FIFO 순서로 처리된다. 완료 메시지의 수령시(상태 블록) 대응하는 ORB가 ORB 리스트로부터 삭제된다.

(2) 새롭게 생성된 ORB가 ORB 리스트의 끝에 부가된다.

(3) ORB 리스트에 링크될 수 있는 ORB의 최대 수는 후술되는 바와 같이 프린터의 2개의 큐(two queues)의 전체 용량과 동일하다.

항목 (3)을 구현하기 위해서는 ORB 관리부(206)는 프린터의 2개의 큐에 응답하는 2개의 카운터를 준비한다. 그중 하나는 CurrentQuedQue라고 하는 카운터로서, 프린터의 프리페치 큐(prefetch queue)(후술됨) 내의 현재의 엠프티 위치의 수를 나타낸다. 다른 카운터는 CurrentImmediateQUE라고 한다. 이 카운터에 대응하는 큐 용량은 본 실시예에서는 1이고, 처리되는 엔트리만이 큐될 수 있다. 이들 카운터의 내용은 ORB의 생성/삭제에 대응하여 증가/감소된다.

호스트 컴퓨터가 ORB를 생성하는 경우에는 DOORBELL 레지스터라고 하는 레지스터에 임의의 값을 기록하여 ORB의 생성을 프린터에 통보할 수 있다. 이러한 절차는 SBP - 2에 규정되어 있으며 그 매뉴얼 등에 설명되어 있다.

HPT 프로세서(203)는 상태 큐(204) 및 큐 프로세서(205)를 포함하고 있다. 1394 인터페이스를 통해 수신된 상태는 상태 식별부(202)에 의해 식별되고, ORB 관리부(206)에 바로 보내지거나 또는 상태 유형에 좌우되는 상태 큐(204)에 부가된다. 상태 큐(204)에 첨부된 상태는 FIFO 순서로 큐 프로세서(205)에 의해 처리된다. 이들 상태에는 2개의 유형이 있다.

(1) 정상 상태... 이 상태는 호스트 컴퓨터 및 프린터 사이에 데이터 전송 결과를 통지하는 상태 블록이며, ORB 관리부(206)에 직접 전송된다.

(2) 언소리사이티드 상태(Unsolicited stats)... 이 상태는 프린터로부터 호스트 컴퓨터로 전송되는 비동기 데이터가 생성되었음을 나타내는 상태 블록이며, 상기 상태 큐(204)에 부가된다. 통상적으로 이 상태는 프린터에 의해 자발적으로(spontaneously) 발행된다.

이들 상태 유형은 상태 블록 내에 기록된 값에 의해 식별된다.

상태 레지스터(210)는 프린터가 데이터를 기록하여 호스트 컴퓨터에 의해 판독될 데이터의 존재를 표시하는 레지스터이다.

이니시에이터의 기능을 하는 호스트 컴퓨터는 상기한 기능적인 구성을 갖고 있다.

<타겟의 구성>

도 1은 타겟의 역할을 하는 프린터의 기능적 구성을 나타내는 블록도이다.

도 1에서 DOORBELL 레지스터(108)는 호스트 컴퓨터에 의해 데이터가 기록되는 레지스터이다. DOORBELL 레지스터 내에 임의의 값을 기록하는 것은 새로운 ORB의 생성을 의미한다. 명령 페치 에이전트(command fetch agent; 103)는 1394 인터페이스(101)를 통해 ORB를 판독하고, 판독 ORB를 프리페치 큐(104)에 첨부하거나 또는 그 유형에 좌우되는 즉시 실행 에이전트(immediate execution agent; 106)에 전송한다. 명령의 유형은 즉시 실행(immediate execution) 또는 순차(순차 실행된) 실행(queued execution)을 나타내는 필드를 참조하여 결정된다. 그러나, 실질적으로는 이러한 필드는 명령의 기능에 대응한다. 예를 들어 본 실시예에서 타겟으로부터 이니시에이터로의 데이터 전송을 위한 명령(리드 요구 명령;REQUESTED READ command)과 타겟 상태를 얻기 위한 명령은 즉시 실행 명령(immediate execution commands)이고, 이니시에이터로부터 타겟으로 프린터 데이터를 전송하기 위한 명령은 순차 실행 명령이다.

순차 실행 에이전트(105) 및 즉시 실행 에이전트(106)는 호스트 컴퓨터의 버퍼로부터 데이터를 판독하거나 또는 명령 페치 에이전트(103)에 의해 판독된 ORB의 내용에 따라 데이터 프로세서(107)로부터 공급된 데이터를 기록한다. 그후, 이러한 에이전트는 호스트 컴퓨터에 정상 상태로 되돌려진다.

더욱이 즉시 실행 에이전트(106)는 PDL, 및 디바이스 관리(device management)를 해석하고 실행함으로써 래스터 데이터를 생성하기 위한 래스터리제이션(rasterization)을 수행하는 데이터 프로세서(107)로부터의 데이터 전송 요구에 응답하여 호스트 컴퓨터에 언소리사이티드 상태(unsolicited status)를 전송한다. 프린터(100)로부터 호스트 컴퓨터(200)의 시스템 메모리(208) 상의 원하는 메모리 위치의 액세스를 위해 버스 인터페이스(102)가 사용된다.

본 실시예에서 순차 실행 에이전트는 호스트 컴퓨터로부터 프린터로의 데이터 전송 ORB를 위해 사용되고, 즉시 실행 에이전트는 프린터로부터 호스트 컴퓨터로의 데이터 전송을 위해 사용된다.

이니시에이터와 타겟의 구성과 동작(arrangement and operation)을 간단히 설명한다. 그 상세한 설명에 앞서 ORB들의 내용을 상세히 설명한다.

<명령 ORB(Operation Request Block)의 내용>

도 3a와 도 3b는 ORB의 일반적인 포맷을 나타낸다. 도 3a에서 "Next_ORB"(링크) 필드(301)는 다음 ORB로의 링크를 저장한다. 다음 ORB가 없다면 그 소정의 값을 나타내는 소정의 값이 저장된다. 제1 ORB는 소정의 어드레스 레지스터에 의해 표시된다. "data_descriptor"(데이터 어드레스) 필드(302)는 데이터 버퍼의 어드레스를 나타낸다. "d"(방향) 필드(303)는 호스트 컴퓨터로부터 프린터로의 데이터 전송(0: 기록) 또는 프린터로부터 호스트 컴퓨터로의 데이터 전송(1: 판독)을 나타낸다. "data_size"(데이터 사이즈) 필드(304)는 어드레스 필드(302)에 의해 표시된 데이터 버퍼의 크기를 나타낸다. 이러한 "Next_ORB" 필드(301) 내지 "data_size" 필드(304)는 SBP-2로 특정되고, 후술될 필드(305 내지 308)가 HPT로의 유일한 처리에 사용된다.

"i" 필드(즉시 비트;immediate bit;305)는 ORB가 즉시 또는 순차 실행 에이전트에 의해 타겟에서 실행될 것인지를 나타낸다. 필드 내의 값이 "0" 즉, 순차 실행 에이전트라면 ORB는 프리페치 큐에 위치하게 되고; 값이 "1"이라면 ORB는 즉시 실행 에이전트에 의해 처리된다. "기능" 필드(function field)(306)는 도 3B에 나타낸 것과 같은 ORB의 의미(meaning)를 나타낸다. 이것은 다음에 설명한다. "command_block_length"(명령 길이) 필드(307)는 "제어 블록"(control block) 필드(308)의 길이를 나타낸다. 제어 블록 필드(308)에는 기능 필드(306)의 값에 대응하여 여러 값이 저장된다.

기능 마다 ORB 의 내용을 설명한다.

(큐 길이 명령;QUEUE DEPTH command)

도 4는 기능=0의 QUEUE DEPTH command을 나타낸다. 이 명령은 타겟의 프리페치 큐(104)의 깊이를 얻기 위해 사용된다. 즉시 비트는 "1"로 설정된다.

이 명령의 제어 블록은 2개의 필드 즉, "source_ID"(소스 ID) 필드(401)와 "status_queue_length"(상태 큐 길이) 필드(402)를 포함하고 있다. 이 소스 ID 필드는 이니시에이터에서 로그된 처리의 식별자를 저장한다. 도 2에 나타낸 실시예에서 로그인 프로세스(logged-in process)는 프린터 드라이버이다. 필드가 부가되어 타겟으로 하여금 복수의 프로세스가 로그인 된 경우 응답하는 프로세스를 확인할 수 있도록 해준다. 상태 큐 깊이 필드(status queue depth field;402)는 상태 큐(204)의 깊이의 타겟을 이니시에이터에 통지한다.

상태 큐 깊이 필드는 상태 큐에서 순차 언소사이티드 상태 블록의 수를 타겟에서 관리하는데 사용된다. 타겟은 이니시에이터에 의한 프리페치 큐의 관리로서 동일한 방식으로 언소사이티드 상태의 생성/처리 완료 메시지에 따라 상태 큐의 깊이를 관리한다.

QUEUE DEPTH 명령의 수신시 타겟은 상태 큐 길이를 카운터 CurrentUnsolicitedQUE에 저장하고, 프리페치 큐 길이를 이니시에이터에 돌려준다. 이니시에이터는 이러한 명령에 의해 얻어진 프리페치 큐의 큐 깊이에 기초하여 ORB들의 생성과 삭제에 응답하여 타겟에 큐되어 있는 ORB들의 개수를 관리한다.

(데이터 전송 명령)

도 5는 기능=1인 데이터 전송 명령을 나타낸다. 이러한 명령은 이니시에이터로부터 타겟으로의 데이터 전송시 사용된다. "page_table_ present" 비트(501)는 페이지 테이블의 존재를 나타낸다. 페이지 테이블이 존재하는 경우에는 페이지 테이블로 참조되는 페이지 사이즈는 "page_size" 필드(502)에 설정된다. 전송될 데이터 버퍼는 이러한 페이지 테이블, 어드레스(503)와 데이터 사이즈(504)에 의해 표시된다.

DATA TRANSFER command의 수신시 타겟은 지정된 데이터 버퍼에 데이터를 기록하거나 또는 방향 필드(direction field)에 따라 그 버퍼로부터 데이터를 출력한다.

(READ 요구 명령;REQUESTED READ command)

도 6은 기능=2의 READ 요구 명령을 나타낸다. 이러한 명령은 타겟으로부터 "READ REQUEST status" 즉, 타겟인 프린터로부터 호스트로 데이터 전송 요구가 전송되어 온 경우에, 데이터가 프린터에 기록되도록하는 데이터 버퍼를 제공한다. "Sequence_number"(시퀀스 번호) 필드(601)는 대응하는 READ REQUEST status에 첨부된 시퀀스 번호와 동일한 값으로 설정되고, 상기 상태는 이러한 명령의 발생에 의해 선동된다. 이러한 값은 REQUESTED READ 상태와 REQUESTED READ 명령이 서로 대응하도록 하는 개수이다. 기타 필드의 포맷은 DATA TRANSFER 명령의 포맷과 동일하다. 이러한 명령에서 즉시 비트는 "1"로 설정되고, 방향 비트 역시 "1"(판독)로 설정된다. 즉시 비트가 "1"로 설정되는 이유는 타겟에 의해 발행된 READ REQUEST 상태에 즉시 응답하기 위해서이다. 버퍼 사이즈는 READ REQUEST 상태에 의해 지정된 값을 이용한다.

DATA TRANSFER 명령의 수신시 타겟은 이니시에이터의 시스템 메모리상에 보증되는 지정된 데이터 버퍼에 데이터를 기록한다.

(직접 상태 응답 명령; DIRECT STATUS RESPONSE command)

도 7은 기능=3인 DIRECT STATUS RESPONSE 명령 ORB를 나타낸다. 이 명령은 이니시에이터가 타겟으로 하여금 판독 요구를 포기하도록 하는 경우 READ REQUEST 상태에 응답하여 발행된다. 대안적으로 이러한 명령은 타겟으로부터의 DIRECT 상태에 응답하여 타겟의 응답으로서 사용된다. "sequence_number"(시퀀스 번호) 필드(701)는 대응하는 READ REQUEST 상태나 DIRECT 상태에 첨부된 시퀀스 번호로서 동일한 값으로 설정된다. 기타 필드의 포맷은 DATA TRANSFER 명령의 포맷과 동일하다. 이러한 명령에서 즉시 비트는 "1"로 설정된다.

DIRECT STATUS RESPONSE 명령의 수신시 타겟은 만일 대응하는 상태 시퀀스 번호를 갖는 READ REQUEST 상태를 발행하였다면 판독 요구를 포기하게 된다.

(디바이스 자원 획득 명령; ACQUIRE DEVICE RESPONCE 명령)

도 8a는 기능=8인 디바이스 자원 획득 명령(ACQUIRE DEVICE RESPONCE) 명령을 나타낸다. "resource_ID"(자원 ID) 필드(801)의 의미는 도 8b에 도시된 바와 같다. "0"은 사용된 디바이스 클래스와 논리적 유닛 특성에 좌우되는 값이다. 이러한 시스템에서 "0"은 디바이스 클래스로서의 프린터와 논리적 유닛 특성으로서의 프린트 서비스를 나타낸다.

이러한 디바이스 자원 획득 명령의 수신시 타겟은 자원 ID에 의해 지정된 자원을 이러한 명령의 송신자(sender)인 이니시에이터에 할당한다.

(디바이스 자원 해방 명령;RELEASE DEVICE RESOURCE command)

도 9는 기능=9인 디바이스 자원 해방 명령 명령을 나타낸다. "resource_ID"(자원 ID) 필드(801)의 의미는 도 8b에 도시된 바와 같다.

디바이스 자원 해방 명령의 수신시 타겟은 자원 ID에 의해 지정된 자원을 해방시킨다.

(기본 디바이스 상태 명령; BASIC DEVICE STATUS command)

도 10은 기능=A(Hex)인 기본 디바이스 상태 명령을 나타낸다.

이러한 명령의 수신시 타겟은 자신이 갖는 상태를 기본 디바이스 상태 블록에 넣어 이니시에이터에 응답하게 된다. 이러한 명령을 발행함으로써 이니시에이터는 프린터 상태를 인식할 수 있다. 프린터에서는 예를 들어 용지 사이즈, 지원되는 에뮬레이션 등의 프린터의 정보와 관련된 여러 종류의 상태 정보가 기초 상태로서 타겟으로부터 다시 전송된다.

＜상태 블록의 내용＞

도 11a와 도 11b는 타겟인 프린터로부터 이니시에이터인 호스트 컴퓨터로 다시 전송되는 상태 블록을 나타낸다. 상태 블록은 상기한 각각의 명령 ORB에 대응하여 준비된다. 상태 블록은 상기 타겟의 순차 실행 에이전트 및 즉시 실행 에이전트에 의해 발행된다.

도 11a에서 "ORB_offset_lo" 필드의 제1 필드는 SBP-2로 특정되고, 상태 등에 대응하는 명령 ORB를 나타내기 위한 필드를 포함하고 있다. "i"(즉시 비트) 필드(1101)는 순차 실행 에이전트와 즉시 실행 에이전트(queued and immediate execution agents)의 어느 것이 이러한 상태를 발행하였는지를 나타낸다. 값이 "0"이면 그 값은 상태가 순차 실행 에이전트에 의해 발행되었음을 나타내게 되고; 값이 "1"이면 상태가 즉시 실행 에이전트에 의해 발행되었음을 나타낸다. "hpt_status"(hpt status) 필드(1102)는 도 11b에 나타낸 것과 같은 상태 블록 유형을 나타낸다. "hpt_status_dependent" 필드(1103)는 hpt 상태에 좌우되는 값으로 주어진다. "status_length"(상태 길이) 필드(1104)는 응답 블록(1105)의 길이를 나타낸다. 유형 마다의 상태 블록을 설명한다.

(QUEUE DEPTH status block)

도 12는 상태=0인 QUEUE DEPTH status를 나타낸다. QUEUE DEPTH status는 QUEUE DEPTH command에 응답하는 타겟으로부터의 응답이고, 타겟은 "prefetch_queue_depth" 필드(1201)의 프리페치 큐(104)의 깊이를 설정하여 이니시에이터에 그 상태를 다시 전송한다. 이러한 상태와 함께 이니시에이터는 프리페치 큐의 크기를 검출할 수 있고, 그 사이즈에 대응하여 생성된 ORB의 개수를 관리한다.

(데이터 전송 상태 블록; DATA TRANSFER status block)

도 13은 hpt=1인 데이터 전송 상태를 나타낸다. 데이터 전송 상태는 데이터 전송 명령에 응답하는 타겟으로부터의 응답이고, 데이터 전송 명령 ORB의 처리의 완료시 타겟에 의해 발행된다. 이러한 상태의 수신시 이니시에이터는 하나의 ORB가 처리되었고 타겟 큐로부터 삭제되었음을 검출할 수 있다.

(READ REQUEST status 블록)

이 상태는 hpt 상태=2(그 포맷은 도시 생략)이고, 응답 블록의 이니시에이터에 의해 보증되는 버퍼 사이즈를 갖는다. 통상적으로 타겟은 명령 ORB에 응답해서가 아니라 자발적으로 이러한 상태를 발행한다. 이니시에이터는 이러한 READ REQUEST 상태에 응답하여 상기한 리드 요구 명령 ORB를 발행한다. 이러한 처리를 위한 절차는 후술된다.

(DIRECT 상태 블록)

이러한 상태는 HPT=3이고(그 포맷은 생략됨) 응답 블록에서 적용 레벨의 상태(즉, 타겟의 데이터 프로세서)를 포함하고 있다. 보다 구체적으로는 적용 레벨의 데이터 교환(exchange)은 통상 ORB와 여기에 링크된 데이터 버퍼를 사용하여 행해진다. 그러나 데이터가 매우 작고 응답 블록의 상한(본 실시예에서는 24 바이트) 내에 있는 경우, 응용 - 레벨 응답은 HPT - 레벨 응답으로 획득된다. 이러한 상태에 응답하여 이니시에이터는 DIRECT STATUS RESPONSE 명령 ORB를 발행한다.

(디바이스 자원 획득; DEVICE RESOURCE ACQUIRE status block)

이 상태는 hpt 상태=8(그 포맷은 도시생략됨)이고, 디바이스 자원 획득 명령을 수신하여 그것을 처리하는 타겟에 의해 발행된다.

(디바이스 자원 해제 상태 블록; DEVICE RESOURCE RELEASE status block)

이 상태는 hpt 상태=9(그 포맷은 도시생략됨)이고, 디바이스 자원 해제 명령을 수신하여 그것을 처리하는 타겟에 의해 발행된다.

(기본 디바이스 상태 블록; BASIC DEVICE status block)

이 상태는 hpt 상태=A(HEX)(그 포맷은 도시생략됨)이고, 기본 디바이스 상태 명령의 수신시 타겟에 의해 발행된다. 이 상태는 소정의 디바이스 상태에 의해 설정된다.

본 실시예의 프린터 시스템에 사용된 명령과 상태 블록을 설명한다. 이니시에이터 및 타겟의 데이터 교환 절차를 설명한다.

<이니시에이터로부터의 데이터 전송 요구 처리; Data Transfer Request Processing from Initiator>

도 14는 호스트 컴퓨터에서의 호스트 컴퓨터로부터 프린터로, 또는 프린터로부터 호스트 컴퓨터로의 데이타 전송 요구에 응답하여 프린터에 명령 ORB의 생성을 통지하기 위한 순서를 나타낸다.

프린터 드라이버 등과 같은 애플리케이션으로부터의 데이타 전송 요구는 단계 S1401에서 모니터된다. 데이타 전송 요구는 프린터 드라이버 등의 애플리케이션으로부터 직접 전송되어야 할 데이타의 존재를 호스트 컴퓨터에 통지하는 것이거나, 또는 프린터로부터의 데이타 판독 요구에 따라 생성되는 것일 수 있다. 프린터가 호스트 컴퓨터와 비동기하여 호스트 컴퓨터로 보내는 상태를 언솔리사이티드 상태(Unsolicited status)라 칭한다. 한편, 호스트 컴퓨터로부터의 명령 ORB의 처리 완료 메시지로서 되돌아오는 상태를 정상 상태(normal status)라 칭한다.

데이타 전송 요구의 검출에 따라, 단계 S1402에서 데이타 전송이 순차(queued) 또는 즉시 실행 명령(immediate execution command)에 의해 실행되는지가 결정된다. 데이타 전송이 즉시 실행 명령에 의해 실행되면, ORB의 즉시 비트(immediate bit)가 설정된다. ORB가 프린터로부터의 언솔리사이티드 상태에 응답하여 발생되면, 데이타 전송은 즉시 실행 명령에 의해 실행되고, ORB가 호스트 컴퓨터의 애플리케이션으로부터의 데이타 전송 요구에 응답하여 발행되면, 데이타 전송은 순차 실행 명령에 의해 실행된다.

순차 실행의 경우, 단계 S1403에서 카운터 CurrentQuedQUE가 "0"인지를 체크한다. 전원 투입이나 리세팅시에, 프린터의 프리페치(prefetch)의 깊이가 QUEUE DEPTH 명령에 의해 판독되어, 카운터 CurrentQuedQUE의 디폴트값으로서 설정된다. 특히, 카운터 CurrentQuedQUE는 현재의 프리페치 큐 내의 빈 위치의 수를 카운트한다. 단계 S1403에서 빈 위치의 수가 "0"인 것으로 결정되면, 타겟의 프리페치 큐가 비어있지 않기 때문에, 큐가 빈 공간이 될 때까지 제어를 대기한다. 빈 위치가 발견되면, 단계 S1404에서 카운터 CurrentQuedQUE의 값을 1만큼 감소시키고, 단계 S1407에서 데이타 전송 ORB를 생성하여 ORB 리스트에 링크한다. 그 후, 단계 S1408에서 임의의 값이 DOORBELL 레지스터에 기록되어, 새로운 ORB의 생성을 타겟에 통지한다.

한편, 단계 S1402에서 데이타 전송이 즉시 실행 명령에 의해 행해진 것으로 결정되면, 단계 S1405에서 카운터 CurrentImmediateQUE의 값이 "0"보다 큰지를 체크한다. 타겟의 즉시 실행 에이전트에 큐가 준비되어 있지 않아, 이 카운터의 최대값은 "1"이다. 따라서, 카운터 CurrentImmediateQUE는 리세팅시 "1"로 설정된다. 카운터 CurrentImmediateQUE의 값이 "0"보다 크면, 단계 S1406에서 카운터의 값을 1만큼 감소시킨다. 그 때, ORB를 ORB 리스트에 링크하여 도어벨을 기록한다.

이러한 방법으로 DOORBELL 레지스터에 기록함에 따라, 페치 에이전트는 도 15에 나타낸 순서로 ORB를 타겟 내로 페치한다.

<페치 에이전트에 의한 처리>

도 15는 DOORBELL 레지스터(108)에서의 기록시에 타겟의 페치 에이전트에 의해 실행되는 처리 순서를 나타낸다.

DOORBELL 레지스터(108)에서의 기록시에, 단계 S1601에서 시스템 메모리 내의 제1 링크 ORB의 어드레스를 판독 포인터로 설정한다.

단계 S1602에서, 판독 포인터에 의해 표시되는 ORB의 즉시 비트를 테스트하여, 즉시 또는 순차 실행 명령인지를 체크한다. ORB가 순차 실행 명령이면, 단계 S1603에서 프리페치 큐(104)의 엔드 어드레스(NextWritePointer)가 획득된다. 호스트 컴퓨터가 프리페치 큐의 빈 위치를 확인한 후에 도어벨을 기록하기 때문에, 큐는 빈 위치를 확실하게 갖는다.

단계 S1604에서 판독 포인터에 의해 표시되는 ORB가 프리페치 큐의 맨 끝에 카피되고, 단계 S1605에서 페치 에이전트가 순차 실행 에이전트에 ORB의 수신을 통지한다. 그 후, 단계 S1606에서, 판독 포인터에 의해 표시되는 ORB의 "Next_ORB" 필드(다음의 ORB로의 링크 필드)가 NULL인지, 즉 링크된 ORB가 존재하는지를 체크한다. 그 필드가 NULL이면 종료하고, 이와 반대이면 그 링크된 ORB의 어드레스를 판독 포인터로 설정하여 단계 S1602부터 반복한다.

한편, 단계 S1602에서 ORB가 즉시 실행 명령인 것으로 결정되면, 단계 S1608로 진행하여, 그 ORB를 즉시 실행 에이전트에 의해 표시되는 어드레스에 카피한다. 그 후, 단계 S1609에서 페치 에이전트가 ORB의 수신을 즉시 실행 에이전트에 통지하고, 단계 S1606으로 진행한다.

이와 같이, ORB는 타겟 내로 페치되어 ORB의 수신을 각 실행 에이전트에 통지한다. 그 후, ORB는 도 16에 도시한 순서로 처리된다.

<실행 에이전트에 의한 처리>

실행 에이전트에는 즉시 및 순차 실행 에이전트가 있는데, 이들은 동일한 순서로 ORB를 처리한다. 따라서, 이들의 처리는 도 16을 이용하여 동시에 설명할 것이다.

실행 에이전트가 페치 에이전트로부터 ORB의 수신을 통지받으면, 단계 S1701에서 Nextreadpointer에 의해 표시되는 ORB의 데이타 어드레스 필드, 방향 비트, 및 데이타 사이즈 필드의 값들을 추출한다. Nextreadpointer는 큐 내에 기억되어 있는 포인터로, 실행 에이전트가 지금 막 처리하려고 하는 ORB를 표시한다. 이 포인터는 순차 실행 에이전트의 경우에 프리페치 큐 내에 제1 ORB를 표시한다.

단계 S1702에서, 추출된 데이타 사이즈가 "0"인지를 체크한다. 그 사이즈가 "0"이 아니면, 단계 S1703에서 방향 비트(Direction bit)가 기록 또는 판독을 표시하는지를 체크한다. 방향 비트가 기록을 표시하면, 실행 에이전트는 데이타 어드레스 및 데이타 사이즈에 의해 지정되는 데이타를 이니시에이터의 시스템 메모리로부터 판독하여, 단계 S1704에서 판독된 데이타를 데이타 프로세서(107)로 전달한다.

데이타 프로세서는 단계 S1707 내지 S1707에서 전달된 데이타를 처리한다. 예를 들면, 프린터의 경우에는 PDL 데이타가 해석(interpret)되고 래스터라이즈(rasterize)된다. 단계 S1707에서, 실행 에이전트는 처리의 종료를 통지한다.

데이타 프로세서에서의 처리 종료를 통지함에 따라, 단계 S1708에서 실행 에이전트는 ORB의 처리 종료를 표시하는 메시지로서 DATA TRANSFER 상태 블럭(정상 상태)을 생성하고, 단계 S1709에서 상태 레지스터(210) 내에 메시지를 기록하여, 상태를 이니시에이터에 통지한다.

한편, 단계 S1703에서 ORB의 방향 비트가 판독을 표시하는 것으로 결정되면, 단계 S1710에서 실행 에이전트가 ORB의 데이타 어드레스 필드 및 데이타 사이즈 필드에 의해 지정되는 데이타 버퍼 내에 데이타를 기록한다. 타겟에 의해 기록된 데이타를 역 데이타라 칭한다. 기록되는 역 데이타는 일반적으로 데이타 프로세서(107) 등에 의해 생성되어 버퍼 내에 기억된다. 예를 들면, 역 데이타는 프린터의 빌트-인(built-in) 폰트 리스트 등의 24 바이트 이상의 데이타이다.

단계 S1711에서, 기록되어진 데이타가 기억된 버퍼가 개방된다.

처리가 종료됨에 따라, 단계 S1713에서 실행 에이전트가 처리 완료 메시지의 상태 블럭(정상 상태)을 생성하고, 단계 S1714에서 상태 레지스터 내에 선정된 값을 기록하여 상태 블럭 메시지를 호스트 컴퓨터에 통지한다.

마지막으로, 단계 S1715에서 실행 에이전트가 카운터 CurrentUnsolicitedQUE의 값을 1만큼 증가시킨다. QUEUE DEPTH 명령의 수신시, 그 "status_que_depth" 필드 내에 기억된 값이 카운터 CurrentUnsolicitedQUE의 디폴트값으로서 설정된다. 이 카운터의 내용은 언솔리사이티드 상태 블럭의 송신 및 그 처리 완료 메시지에 따라 증가/감소되어, 상태 큐(204)의 빈 위치의 수를 표시한다. 카운터 CurrentUnsolicitedQUE는 다음과 같은 조건 하에서 카운트된다.

- 카운트다운 조건

(1) 언솔리사이티드 상태 블럭의 송신시, 카운터는 1만큼 감소된다. 이니시에이터에서, 언솔리사이티드 상태만이 상태 큐(204)에 큐잉된다.

- 카운트업 조건

(1) REQUESTED READ ORB의 처리 완료 상태 블럭의 송신시, 카운터는 1만큼 증가된다. 이 상태의 수신시, 이니시에이터는 상태 큐(204)로부터 처리된 상태 블럭을 제거한다.

(2) 데이타 사이즈 = 0인 ORB의 처리 완료 상태 블럭의 송신시, 카운터는 1만큼 증가된다. 한편, DIRECT STATUS RESPONSE ORB의 상태 블럭의 송신시, 카운터는 1만큼 증가된다. 이 상태의 수신시, 이니시에이터는 상태 큐(204)로부터 처리된 상태 블럭(다이렉트 상태)을 제거한다.

단계 S1702에서 데이타 사이즈가 "0"인 것으로 결정되면, 처리해야할 데이타가 없으므로, 단계 S1713에서 실행 에이전트는 어떠한 처리없이 상태 블럭으로 복귀하여 처리 완료 메시지를 보낸다.

상술한 순서에 의해, 이니시에이터에 의한 ORB의 생성, 및 타겟에 의한 그 처리 및 상태 블럭의 생성이 행해진다. 이하, 도 17 및 도 18을 참조하여 프린터로부터 호스트 컴퓨터로 송신된 언솔리사이티드 상태에 따른 데이타 송신에 대해 설명한다.

<타겟에서의 언솔리사이티드 상태의 발행>

예를 들어 용지없음(out-of-paper), 잼(jam)과 같은 에러가 프린터에 발생되면, 프린터로부터 호스트 컴퓨터로 몇 종류의 정보가 즉시 통지되어야 한다. 이러한 경우, 프린터는 호스트 컴퓨터로부터의 명령에 비동기하여 호스트 컴퓨터에 자발적으로 데이타를 전송한다. 프린터로부터 보내야할 데이타가 발생되면, 데이타 프로세서는 상기 데이타의 존재를 즉시 실행 에이전트에 통지하여, 도 17에서의 처리를 개시한다.

단계 S1801에서 호스트 컴퓨터로 전송되어야 할 데이타가 24 바이트를 초과하는지를 체크한다. 24 바이트는 상태 블럭의 응답 블럭 내에 기억될 수 있는 데이타 량의 상한이다.

데이타가 상한값을 초과하면, 단계 S1802에서 READ REQUEST 상태 블럭(언솔리사이티드 상태)을 생성하고, 단계 S1803에서 상기 상태 블럭의 "ORB_offset"에 역 데이타의 기억 어드레스를 설정한다. 이 때, 역 데이타의 사이즈도 상태 블럭 내에 기록된다.

단계 S1804에서 카운터 CurrentUnsolicitedQUE의 값이 "0"보다 큰지, 즉 상태 큐가 빈 위치를 갖는지를 체크한다. 단계 S1804에서 "아니오"이면, 큐가 빈 위치를 가질 때까지 제어를 대기한 후, 단계 S1805에서 카운터 CurrentUnsolicitedQUE의 값을 1만큼 감소시켜, 단계 S1806에서 상태 레지스터에 적절한 값을 기록한다.

한편, 데이타가 24 바이트 이하이면, 데이타는 상태 블럭 내에 캡슐화되어 있는 동안에 호스트 컴퓨터로 보내질 수 있으므로, DIRECT 상태(언솔리사이티드 상태)를 생성하여, 그 명령 세트 의존 필드(Command set dependent field) 내에 역 데이타를 기억한다. 그 후, 단계 S1804로 분기한다.

이와 같이, 언솔리사이티드 상태가 이니시에이터로 보내진다.

상태를 수신하면, 이니시에이터는 도 18에 나타낸 순서로 그것을 처리한다.

<이니시에이터에 의한 상태 처리>

이 처리는 상태 레지스터에 선정된 값을 설정함으로써 생기는 인터럽트에 의해 개시된다. 인터럽트는 상태 레지스터에 값이 설정된 때에 발생되며, 다중 인터럽트를 허가하고 있다.

단계 S1501에서, 상태가 정상 또는 언솔리사이티드 상태인지를 체크한다. 상태가 언솔리사이티드 상태이면, 단계 S1502에서 타겟으로부터 상태 블럭을 판독하여, 상태 큐의 맨 끝에 카피한다. 단계 S1503에서, 그 블럭이 큐의 선두 위치로 이동되었는지를 체크한다. 단계 S1503에서 "예"이면, 단계 S1504에서 역 데이타를 기억하기 위한 버퍼를 확보하고, 단계 S1505에서 상태가 READ REQUEST 또는 DIRECT 상태인지를 체크한다.

상태가 READ REQUEST 상태이면, 단계 S1506에서 REQUESTED READ 명령 ORB가 생성된다. 생성된 ORB를 "즉시" 플래그로 설정하고, 그 방향 플래그는 판독을 표시한다. 또한, 판독되어야 할 데이타를 기억하는 버퍼의 어드레스는 ORB 내에 기록된다. 그 후, 단계 S1507에서 데이타 전송을 요구하는 이벤트가 발생된 것을 도 14에 나타낸 프로세스에 통지한다. 도 14에 나타낸 프로세스는 단계 S1506에서 생성된 ORB를 타겟으로 보내고, 타겟으로부터의 데이타가 데이타 버퍼 내에 기록되어, 데이타 전송을 달성하게 된다.

단계 S1505에서 상태가 DIRECT 상태인 것으로 결정되면, 단계 S1508에서 그 상태 내의 역 데이타가 확보된 버퍼에 카피되고, 단계 S1509에서 애플리케이션 등의 호스트 프로세스로 보내진다. 그 후, DIRECT STATUS RESPONSE ORB (즉시 비트가 "즉시"로 설정됨)가 단계 S1510에서 생성된다. 이때, 데이타 사이즈 = 0으로 지정된다.

한편, 단계 S1501에서 상태가 정상 상태인 것으로 결정되면, 단계 S1511에서 그 상태에 대응하는 ORB의 방향 비트가 판독 또는 기록을 나타내는지를 체크한다. 방향 비트가 기록을 나타내면, 즉 이니시에이터로부터 타겟으로의 데이타 전송을 나타내면, 대응하는 ORB가 리스트에서 삭제되고(단계 S1512), 그 ORB에 의해 이용된 데이타 버퍼가 개방되며(단계 S1513), 프리페치 큐의 빈 사이즈를 나타내는 카운터 CurrentQuedQUE의 값이 1만큼 증가된다(단계 S1514).

한편, 방향 비트가 판독을 나타내면, 즉 타겟으로부터 이니시에이터로의 데이타 전송을 나타내면, 그 상태가 REQUESTED READ COMMAND ORB에 대한 응답이므로, 단계 S1515에서 프린터 드라이버 등의 호스트 프로세스에 판독의 종료를 통지한다. 단계 S1516에서 대응하는 ORB가 리스트에서 삭제되고, 카운터 CurrentImmediateQUE가 1만큼 증가된다. 즉, 즉시 실행 에이전트는 다음의 ORB를 송신하기 위한 상태가 된다.

이와 같이, 상태 레지스터에 기록시에, 이니시에이터는 상태를 처리한다. 즉, 언솔리사이티드 상태는 큐잉되어 처리되지만, 정상 상태는 즉시 처리된다. 언솔리사이티드 상태가 큐잉되는 이유는 ORB를 생성하기 위해서이다. ORB 리스트에 생성되고 링크되는 ORB의 수는 타겟의 실행 에이전트의 처리 용량과 프리페치 큐의 사이즈의 합계 이하로 한정된다. 이와 같이 ORB의 수를 한정하는 것은 즉시 실행되어야 할 ORB가 ORB 생성 메시지 직후에 처리되는 것을 보증하기 위해서이다. 즉시 실행 명령은 타겟으로 전달되는 동안 즉시 실행 에이전트에 의해 즉시 처리된다. 그러나, ORB가 무제한으로 생성되면, ORB 리스트 자체가 불필요하게 즉시 실행 에이전트에 의해 처리되어야 할 ORB의 큐로 되고, 생성된 ORB가 타겟으로 전달되지 않게 된다. 그 결과, 즉시 실행이 보증되지 않게 된다. 링크된 ORB의 수가 타겟의 프리페치 큐 내에 배치된 ORB의 수와, 실행 에이전트에 의해 실행되는 ORB의 수의 전체 값으로 한정되므로, 이니시에이터에 의해 생성되는 ORB가 타겟으로 즉시 전달될 수 있다. 이 때문에, 언솔리사이티드 상태를 수신하더라도, 이니시에이터는 ORB의 수의 한계에 의해 새로운 ORB를 자주 생성할 수 없다. 따라서, 언솔리사이티드 상태는 상태 큐 내에 임시로 배치된다.

＜메시지 시퀀스의 예＞

이하, 상술한 순서로 이니시에이터와 타겟 사이에서 교환되는 메시지 시퀀스의 일례를 도 19 내지 도 21을 참조하여 설명한다.

(타겟으로의 데이타 전송 시퀀스)

도 19는 이니시에이터(호스트 컴퓨터)로부터 타겟(프린터)으로의 데이타 전송 시의 시퀀스의 예를 나타낸다.

도 19의 SBP-2는 SBP-2 규격에 의해 규정된 데이타, 즉 도 3a 및 3b에 나타낸 명령 및 도 11a 및 11b에 나타낸 상태에서의 SBP-2로 규정된 필드를 처리하는 처리층을 나타낸다. 또한, HPT는 SBP-2로 규정되지 않은 기능마다 정의되는 처리를 행하는 처리층을 나타낸다. HPT는 상술한 플로우차트의 순서를 실행한다. SBP-2는 ORB의 링크, 도어벨의 기록, ORB의 전달 등의 기능을 담당한다.

이니시에이터 상에서의 애플리케이션은 데이타를 생성하고, 그 HPT는 ORB (이 경우, 데이타 전송 ORB)를 생성하며, 빈 큐 카운터(CurrentQueQUE)의 값이 1만큼 감소되어, ORB 링크 요구가 SBP-2에 발행된다(1901). SBP-2는 생성된 ORB를 리스트에 링크하며, DOORBELL 레지스스터에 기록 요구를 발행한다(1902). 1394 인터페이스는 DOORBELL 레지스터 내의 도어벨을 기록하고(1903), 그 후 타겟의 SBP-2가 그 메시지를 수신한다(1904).

메시지의 수신시, SBP-2는 ORB 판독 요구를 1394 인터페이스에 발행하고(1905), 시스템 메모리로부터 ORB가 판독된다(1906). HPT는 그 내용에 따라 대응하는 큐 내에 판독된 ORB를 기억시킨다(1907). 이 경우, ORB가 데이타 전송 명령 ORB이므로, 데이타 판독 요구가 1394 인터페이스에 발행된다(1908). 이 요구에 응답하여, 지정된 어드레스로부터 데이타가 판독되어, 애플리케이션으로 전달된다(1909). 이 경우의 애플리케이션은, 예를 들면 이미지를 매핑(mapping)하기 위한 래스터라이저이며, 프린터 엔진에 의해 매핑된 이미지가 인쇄 출력된다.

데이타의 판독 후, 타겟의 SBP-2에 상태 블럭 송신 요구가 발행되고(1910), 상태 블럭(데이타 전송 상태)이 이니시에이터로 되돌려진다.(1911). 상태의 수신시, 이니시에이터의 HPT가 링크로부터 대응하는 ORB를 삭제하고, 빈 큐 카운터(CurrentQueQUE)를 1만큼 증가시킨다(1912).

상술한 순서에 따라, 이니시에이터로부터 타겟으로 데이타가 전송된다.

(이니시에이터로의 데이타 전송 시퀀스)

도 20은 판독 요구 상태 시퀀스를 나타낸다.

타겟 상의 애플리케이션이 데이타를 생성하면, 언솔리사이티드 상태(판독 요구 상태)가 생성되고, 상태 큐 카운터(CurrentUnsolicitedQUE)가 1만큼 감소되며, 언솔리사이티드 상태 송신 요구가 발행된다(2001). 이 요구의 수신시, SBP-2는 언솔리사이티드 상태 블럭을 이니시에이터(2002)로 송신한다(2002). 이 상태의 수신시, 이니시에이터의 SBP-2층은 언솔리사이티드 상태 블럭을 HPT 층으로 송신한다(2003). HPT 층은 언솔리사이티드 상태 블럭을 상태 큐에 카피한다.

상태 큐의 선두 위치에서의 상태에 대하여, HPT층은 데이타 버퍼 및 판독 요구 ORB를 준비하여, SBP-2 층에 생성된 ORB의 링크를 요구한다(2004). SBP-2 층은 DOORBELL 레지스터에 기록 요구를 발행하고(2005), 1394 인터페이스는 도어벨을 기록한다(2006). 도어벨 기록 메시지가 1394 인터페이스로부터 SBP-2로 보내지고(2007), ORB의 판독 요구의 발행(2008) 및 ORB의 판독(2009)이 즉시 행해진다. 판독 ORB는 타겟의 HPT 층으로 전달되고, HPT 층은 지정된 큐 내의 그 ORB를 기억한다(2010). 그 후, HPT 층은 ORB의 내용을 해석하여 데이타 판독 요구를 인식하고, 1394 인터페이스에 데이타 기록 요구를 발행한다(2011).

이 요구에 응답하여, 이니시에이터에 의해 준비된 데이타 버퍼 내에 역 데이타가 기록된다(2012).

이 처리의 완료시, 타겟은 상태 블럭 생성 요구를 발행하고, 상태 큐의 빈 위치의 수를 1만큼 증가시킨다(2013). 이 요구에 응답하여, 정상 상태 블럭이 이니시에이터로 보내진다(2014). 그 상태에 응답하여 HPT에 ORB 링크 삭제 요구가 발행되면, 대응하는 ORB가 삭제되고, 즉시 실행 에이전트의 큐의 빈 위치의 수("0"이어야 함)가 1만큼 증가되어, 사용된 데이타 버퍼를 개방한다(2015).

이 순서에 따라, 타겟으로부터 이니시에이터로 데이타가 전송된다.

(이니시에이터로의 데이타 전송 순서)

도 21은 타겟으로부터 이니시에이터로의 데이타 전송시의 순서를 나타낸다. 도 20의 순서와는 달리, 전송되어야 할 데이타가 작기 때문에 DIRECT 상태가 사용된다.

타겟 상의 애플리케이션이 데이타를 생성하면, 언솔리사이티드 상태(DIRECT 상태)가 생성되고, 상태 큐 카운터(CurrentUnsolicitedQUE)가 1만큼 감소되며, 언솔리사이티드 상태 송신 요구가 발행된다(2101). 이 요구의 수신시, SBP-2는 언솔리사이티1드 상태 블럭을 이니시에이터로 송신한다(2102). 이 상태의 수신시, 이니시에이터의 SBP-2 층은 언솔리사이티드 상태 블럭을 HPT 층으로 송신한다(2103). HPT 층은 언솔리사이티드 상태 블럭을 상태 큐를 카피한다.

상태 큐의 선두 위치에서의 상태에 대하여, HPT 층은 그 상태 내에 캡슐화된 애플리케이션 레벨 상의 데이타를 판독하고, 데이타 버퍼 및 DIRECT 상태 응답 ORB를 준비하며, SBP-2 층에 생성된 ORB의 링크를 요구한다(2104). SBP-

2 층은 DOORBELL 레지스터에 기록 요구를 발행하고(2105), 1394 인터페이스는 도어벨을 기록한다(2106). 도어벨 기록 메시지는 1394 인터페이스로부터 SBP-2로 보내지고(2107), ORB의 판독 요구의 발행(2108) 및 ORB의 판독(2109)이 즉시 행해진다.

판독 ORB가 타겟의 HPT 층으로 전달되고, HPT층은 지정된 큐 내에 그것을 기억한다(2110). 그 후, HPT 층은 ORB의 내용을 해석한다. ORB가 DIRECT 상태 응답 ORB인 것으로 확인되면, 타겟은 대응하는 상태 블럭 생성 요구를 발행하고, 상태 큐의 빈 위치의 수를 1만큼 증가시킨다(2111). 이 요구에 응답하여, 정상 상태 블럭은 이니시에이터로 되돌려진다(2112). ORB 링크 삭제 요구가 그 상태에 응답하여 HPT에 발행되면, 대응하는 ORB가 삭제되고, 즉시 실행 에이전트의 큐의 빈 위치의 수가 1만큼 증가되어, 사용된 데이타 버퍼가 개방된다(2113).

이 순서에 따라, 타겟으로부터 이니시에이터로 데이타가 전송된다. 이 순서에서, 도 20에 나타낸 것과 비교하여, 2 단계, 즉 데이타 기록 요구(2011) 및 역 데이타(2012)의 기록이 생략되어 있다.

〈본 시스템의 고유 기능〉

본 시스템의 구성 및 동작은 상술한 바와 같다. 본 시스템에 고유한 상기 기능, 구성, 장점들은 이하에서 요약된다:

(1) 두개의 실행 에이전트 예를 들어, 큐잉 및 즉시 실행 에이전트가 타겟에 서 준비되고, 명령 큐들은 이 실행 에이전트와 대응하여 제공된다. 이러한 방법으로, 이니시에이터로부터의 데이타 전송 요청(ORB)이 큐잉되고 순서대로 처리되나(순차 실행), 타겟으로부터의 READ REQUEST 상태는 대응하는 REQUESTED READ ORB가 발행된 다음 즉시 처리된다(즉시 실행). ORB 리스트에 링크된 ORB의 수의 상한선이 기록 ORB에 대해서는 순차 실행 큐의 크기에 제한되고, 판독 ORB에 대해서는 즉시 실행 큐의 크기(이 경우 1)에 제한되기 때문에 즉시 실행이 보장된다.

한편, 타겟으로부터의 READ REQUEST 상태는 이니시에이터 내에서 큐잉된다. 이러한 이유로, 이니시에이터로부터의 데이타 전송 요청은 프리페치 큐에 부가되고 타겟으로부터의 데이타 전송 요청(판독 요청)은 상태 큐에 부가된다. 이러한 제어로서, 이니시에이터와 타겟 상의 애플리케이션 사이에 전이중 통신 채널이 제공될 수 있다. 즉, 데이타 전송 방향에 관계없이 큐에 부가된 요청은 FIFO 순차로 처리되어, 단방향으로의 데이타 전송이 다른 방행으로의 전송에 영향을 미치지 않음으로써, 상호 간섭하지 않는 독립적인 채널을 제공한다. 다시 말해서, 이니시에이터 또는 카겟에서 발생된 데이타는 이니시에이터로부터 타겟 또는 그 반대로 비동기적으로 전송될 수 있다.

(2) 이니시에이터 및 타겟이 서로의 큐의 빈 공간 크기를 감시하기 때문에, 전송된 ORB 또는 상태 블럭이 확실히 수신된다.

(3) 전이중 통신 채널이 싱글 로그인 프로세스에 의해 제공된다. 즉, 이니시에이터로서 많은 리소스를 갖는 컴퓨터를 사용하고 타겟으로서 불량한 리소스를 갖는 프린터를 사용하여 상호 데이타가 교환될 수 있기 때문에 특히, 필요한 메모리 용량을 억제할 수 있다.

(4) IEEE1394 인터페이스를 사용하기 때문에, 타겟이 자신의 리소스가 이용가능해진 경우 타겟으로서의 데이타 전송이 수행될 수 있고, 이니시에이터는 타겟의 편의대로 데이타 전송에 의해 점유되는 것을 방지할 수 있다. 만약 프린터가 타겟으로서 역할한다면, 자신의 리소스가 이용가능해진 경우 컴퓨터로부터 통과된 데이타를 판독할 수 있다. 이러한 이유로, 호스트 컴퓨터는 프린터가 데이타를 수신하기 용이하다고 확신한 다음 프린터를 감시하고 데이타 전송을 개시하기 위한 처리를 실행할 필요는 없다. 즉, 호스트 컴퓨터는 프린터 상태에 관계없이 데이타를 프린터에 송신할 수 있고,

프린터가 이용가능해진 후에는 데이타를 전송할 필요없다.

(5) SBP-2를 사용하기 때문에, ORB 하나만이 타겟 내에서 큐잉되고, 자신의 전송될 데이타는 처리 대기 시간 내에 이니시에이터 내에 저장된다. 이러한 이유로, 타겟의 메모리 리소스가 최소화될 수 있다.

(6) DIRECT 상태를 사용하여, 애플리케이션층의 데이타는 HPT층의 상태로 캡슐화될 수 있고, 타겟으로부터 이니시에이터로 전송될 수 있다. 이러한 이유로, 데이타 전송 시퀀스가 단축될 수 있다.

본 실시예의 시스템은 호스트 컴퓨터 및 프린터에만 제한된 것은 아니고 다양한 다른 기기에도 적용될 수 있음을 주지하기 바란다. 상술한 특성들은 호스트 컴퓨터와 프린터 간의 관계 뿐아니라 호스트 컴퓨터와 작은 리소스를 갖는 주변기기 및 주변 기기들간의 접속에도 유용하다.

[제2 실시예]

본 발명의 제2 실시예로서 복수의 논리 채널을 동시에 제공하는 시스템을 설명하고자 한다. 이 경우에서, 타겟측에서 프린터, 팩시밀리, 및 화상 스캐너를 복합하는 디지탈 다중 기능 머쉰이 프린터 자체를 대신하여 사용될 수 있다.

〈시스템 구성〉

도 23은 본 실시예의 이니시에이터의 블럭도이다. 도 23에서, 도 2와의 차이점만을 설명할 것이고, 공통된 요소들의 설명은 생략할 것이다. 도 23의 특징적 사항은 상태 블럭(212)이 상태 유형뿐 아니라 논리 채널을 식별하고 채널 단위로 상태 블럭을 분배하고, 복수개의 채널에 대해 채널당 시스템 메모리와 HPT 프로세서를 갖는 구성을 가진다는 데 있다.

상태 식별자(212)는 상태 블럭에 포함된 채널 ID에 따라 채널을 식별함을 주지하기 바란다. 이는 다음에 설명할 것이다. 예를 들어, 디지탈 다기능 머신이 사용되는 경우, 채널 단위마다 상이한 애플리케이션이 부여될 수 있는데 즉, 1 채널이 프린터 드라이버에 의해 사용되고 다른 채널이 화상 스캐너 드라이버에 의해 사용될 수 있거나, 또는 싱글 애플리케이션이 복수의 채널을 사용할 수 있다.

도 22는 타겟의 블럭도이다. 도 1과의 차이점은 명령 페치 에이전트(113)가 채널 단위로 명령 ORB를 분배한다는 데 있다. 각 채널은 도 1에서와 동일한 구성을 갖고, 프리페치 큐와 두개의 실행 에이전트(큐 및 즉시 실행 에이전트)를 갖는다.

ORB가 논리 채널을 나타내고, 명령 페치 에이전트는 이를 참조하여 채널을 식별한다.

〈명령 ORB의 형식〉

도 24 내지 32B는 명령 ORB 형식들의 예들을 도시한다. 도 24는 데이타 전송 명령 ORB를 도시하고, 도 25는 REQUEST READ 명령 ORB를 도시하고, 도 26은 DIRECT STATUS RESPONSE 명령 ORB를 도시하고, 도 27a 및 28b는 ACQUIRE DEVICE RESOURCE 명령 ORB, 도 28은 RELEASE DEVICE RESOURCE 명령 ORB를 도시하고, 도 30은 BASIC DEVICE STATUS 명령 ORB를 도시한다. 이 명령 ORB들은 이들이 채널 ID 필드를 갖는다는 것을 제외하고는 실질적으로 제1 실시예서와 동일하다.

도 29는 ABDICATE DEVICE RESOURCE RESPONSE 명령 ORB를 도시하는 것으로, 이는 타겟으로부터의 ABDICATE DEVICE RESOURCE 요청에 대한 응답이다. 타겟이 불량한 리소스를 갖기 때문에, 복수개의 채널들이 사용되고 복수개의 애플리케이션 프로세스가 실행되는 경우 프로세스의 리소스가 부족해질 수 있다. 이러한 경우에서, 타겟은 ABDICATE DEVICE RESOURCE 요청을 발행한다.

도 31a는 채널 개방 요청을 발행하기 위해 사용된 OPEN CHANNEL REQUEST 명령이고, 도 31b는 그 명령에 대응하는 OPEN CHANNEL REQUEST 상태를 도시한다. 이 명령 및 상태에 따라, 원하는 논리 채널을 오픈한다.

도 32a는 채널 채널 클로즈 요청을 발행하기 위해 사용된 CLOSE CHANNEL REQUEST를 도시하고, 도 32b는 그 명령에 대응하는 CLOSE CHANNEL REQUEST 상태를 도시한다. 이 명령 및 상태에 따라서, 원하는 논리 채널을 클로즈한다.

〈상태 블럭의 형식〉

도 33a 내지 38는 상태 블럭의 형식을 도시한다. 도 33a 내지 33c는 일반적인 상태 형식을 도시하고, 도 34는 READ REQUEST 상태를 도시하고, 도 35는 DIRECT 상태를 도시하고, 도 36은 ACQUIRE DEVICE RESOURCE 상태를 도시하고, 도 38은 BASIC DEVICE 상태를 도시한다. 이 상태 블럭의 형식은 채널 ID를 포함하는 것을 제외하고는 제1 실시예에서와 실질적으로 동일하다.

도 37은 ABDICATE DEVICE RESOURCE 상태 블럭을 도시하는 것으로, 리소스 ID를 저장하여 이니시에이터에 대하여 ID 필드(3701) 내에서 포기를 요청하도록 사용된다. 이 상태는 타겟에 의해 발생된 요청이기 때문에, 언솔리사이티드 상태로서 이니시에이터에게로 전송된다.

〈이니시에이터 및 타겟에서의 명령/상태 처리 절차〉

도 39a 및 39b는 본 실시예의 이니시에이터 및 타겟에서의 처리 절차를 도시한다. 도 39a는 이니시에이터 내의 상태 레지스터 내에 기록시 개시되고, 제1 실시예의 도 18에 대응한다. 처리가 개시되면, 단계 S3901에서 채널 ID를 식별한다. 그 다음, 단계 S1501 및 도 18에서의 연속하는 단계와 같이 동일한 처리가 식별된 채널에 대하여 실행된다.

도 39b는 타겟에서의 DOORBELL 레지스터 내에 기록시 개시된 처리 절차를 도시하고, 제1 실시예에서의 도 15에 대응한다. 처리가 개시된 경우, 채널 ID가 단계 S3911에서 식별된다. 그 다음, 단계 S1601 및 도 15에서의 후속하는 단계에서와 동일한 처리가 식별된 채널에 대해 실행된다.

더우기, 데이타 전송 요청(도 14)에 응답하여 개시하는 처리, 타겟 에이전트(도 16)에 의한 처리, 및 데이타 발생시 프린터로부터 호스트 컴퓨터에 전송되는 처리는 제1 실시예와 동일하다. 그러나, 이러한 처리 이전에 채널이 오픈되어야 한다.

또한, 타겟으로부터 ABDICATE RESOURCE 요청은 타겟으로부터의 DIRECT 상태를 사용하는 데이타 전송에서와 동일한 시퀀스로 발행된다.

본 실시예에서, 데이타 전송은 상술한 절차에 의해 실행된다. 본 실시예의 시스템은 복수의 논리 채널이 사용될 수 있는 것을 제외하고는 제1 실시예에서와 실질적으로 동일하다. 이러한 이유로, 디지탈 다기능 머쉰 등의 복수의 기기들을 갖는 장비에도 양방향 통신이 제공될 수 있다. 따라서, 제1 실시예에 기술된 기능 (1) 내지 (6)을 복수의 채널에 대해 제공될 수 있다.

[다른 실시예들]

본 발명은 (예를 들어, 호스트 컴퓨터, 인터페이스 기기, 판독기, 프린터 등)의 복수의 장비로 구성된 시스템 또는 (예를 들어, 복사기, 팩시밀리 기기 등의)단일 장비로 구성된 장치에 적용될 수 있다.

또한 본 발명의 목적은 상술한 실시예들의 기능을 실현할 수 있는 소프트웨어 프로그램의 프로그램 코드(즉, 도 14 내지 18 및 도 39a 및 39b에 도시된 프로그램 절차)를 시스템 또는 장치에 기록하는 기억 매체를 제공함으로써 달성될 수 있고, 기억 매체에 저장된 프로그램 코드를 시스템 또는 장치의 컴퓨터에 의해 판독 및 실행한다.

이 경우에서, 기억 매체로부터 판독된 프로그램 코드 자체는 상술한 실시예의 기능을 실행하고, 상기 프로그램 코드를 저장하는 기억 매체가 본 발명을 구성한다.

프로그램 코드를 제공하기 위한 기억 매체로서는 예를 들어, 플로피 디스크, 하드 디스크, 광학 디스크, 자기 광학 디스크, CD - ROM, CD - R, 자기 테이프, 비휘발성 메모리 카드, ROM 등이 사용될 수 있다.

발명의 효과

상술한 실시예들의 기능은 컴퓨터에 의해 판독 프로그램 코드를 실행할 뿐아니라 프로그램 코드의 지시를 기초로 하여 컴퓨터 상에 실행되고 있는 OS(operating system)에 의해 실행되는 임의 또는 실제 처리 동작에 의해 실현될 수 있다.

또한, 상술한 실시예들의 기능은 기억 매체로부터 판독된 프로그램 코드가 확장 보드 등의 메모리 내에 기록된 다음, 컴퓨터 내에 삽입되거나 접속되어 있는 기능 확장 보드 또는 기능 확장 유닛으로 구성된 CPU 등에 의해 샐행된 임의 또는 모든 실제 처리 동작에 의해 실현될 수 있다.

다시 말해서 본 발명에 따르면, 로그인 프로세스, 및 데이타 교환을 위해 필요한 프로세스, 메모리 등의 리소스에 의해 실현될 수 있는 비동기이고 양방향 통신을 가능케하는 전이중 통신이 효율적으로 사용된다.

이니시에이터 및 타겟이 상호 각자의 큐의 빈 공간 크기를 감시하기 때문에, 전송된 ORB 또는 상태 블럭을 용이하게 수신할 수 있다

IEEE1394 인터페이스가 사용되기 때문에, 타겟이 자신의 리소스가 이용가능해져 데이타를 판독하는 경우 타겟으로의 데이타 전송이 수행될 수 있고, 타겟의 편의대로 데이타 전송에 의해 이니시에이터가 점유되는 것을 방지할 수 있다.

SBP - 2가 사용되기 때문에, ORB만이 타겟 내에서 큐잉되고, 전송될 자신의 데이타는 처리 대기 시간 내에 이니시에이터 내에 저장된다. 이러한 이유로, 타겟의 메모리 리소스는 최소화될 수 있다.

DIRECT 상태를 사용하여, 애플리케이션층의 데이타를 HPT층의 상태 내에 캡슐화시켜 타겟으로부터 이니시에이터로 전송될 수 있다. 이러한 이유로, 데이타 전송 시퀀스가 단축될 수 있다.

또한, 다중 채널이 실현될 수 있다.

본 발명의 사상 및 범주를 벗어나지 않는 범위 내에서 많은 광범위한 다른 실시예들이 이루어질 수 있는 바와 같이, 본 발명은 특정한 실시예에 제한되는 것이 아니라 첨부된 청구항으로 제한된다.

(57) 청구의 범위

청구항 1.

이니시에이터(initiator)의 기억 영역 액세스시 타겟(target)으로부터 데이타를 교환하는 통신 제어 방법에 있어서,

상기 이니시에이터는 상기 기억 영역에 대한 판독 및 기록 액세스에 대응하는 명령을 상기 타겟에 의해 보유될 수 있는 판독 및 기록 명령의 수를 초과하지 않도록 상기 타겟으로 송신하며,

상기 타겟은 상기 수신된 판독 및 기록 명령을 상이한 큐(queues) 내에 보유하고 상기 보유된 명령을 독립적으로 처리하는 것을 특징으로 하는 통신 제어 방법.

청구항 2.

제1항에 있어서, 상기 타겟은 상기 기록 명령을 1의 용량을 갖는 큐 내에 보유하고 상기 큐 내에 있는 상기 명령을 처리하는 것을 특징으로 하는 통신 제어 방법.

청구항 3.

제1항에 있어서, 상기 타겟으로부터 상기 기억 영역 내로의 데이타 기록 요청 수신시 상기 이니시에이터는 상기 요청을 제2 큐 내에 보유하고, 기록 명령을 상기 타겟에 발행(issue)하는 것을 특징으로 하는 통신 제어 방법.

청구항 4.

제3항에 있어서, 상기 타겟은 상기 제2 큐에 의해 보유될 수 있는 요청의 수를 초과하지 않도록 상기 이니시에이터에 대하여 기록 요청을 발행하는 것을 특징으로 하는 통신 제어 방법.

청구항 5.

제1항에 있어서, 상기 이니시에이터는 상기 타겟으로부터의 요청에 응답하여 상기 기억 영역 내에 기록 명령을 발행하는 것을 특징으로 하는 통신 제어 방법.

청구항 6.

제5항에 있어서, 상기 이니시에이터에게로 전송될 데이타가 선정된 크기보다 작은 경우, 상기 타겟은 상기 기억 영역 내에 기록할 것을 상기 이니시에이터에 대하여 요청하는 것이 아니라, 상기 이니시에이터 내의 상기 기억 영역의 선정된 영역 내에 상기 데이타를 직접 기록하는 것을 특징으로 하는 통신 제어 방법.

청구항 7.

이니시에이터의 기억 영역 액세스시 타겟으로부터 데이타를 교환하는 통신 제어 방법에 있어서,

상기 타겟은 전송될 데이타의 크기가 선정된 크기를 초과하여 전송되는지 체크하여, 전송될 데이타의 크기가 상기 선정된 크기를 초과하는 경우 상기 기억 영역 내에 기록 명령을 발행하도록 상기 이니시에이터에 대해 요청하며, 전송될 상기 데이타의 크기가 상기 선정된 크기를 초과하지 않는 경우 상기 데이타를 상기 이니시에이터에 송신하며,

상기 이니시에이터는 상기 타겟으로부터 기록 명령 발행 요청 수신시 기록 명령을 발행하는 것을 특징으로 하는 통신 제어 방법.

청구항 8.

제1항에 있어서, 상기 타겟 및 상기 이니시에이터는 채널 식별자(channel identifiers)를 상기 기록 및 판독 명령에 부여하고, 채널 식별자 유닛들 마다 상기 명령들을 독립적으로 처리하는 것을 특징으로 하는 통신 제어 방법.

청구항 9.

이니시에이터의 기억 영역 액세스시 타겟으로부터 데이타를 교환하기 위한 통신 시스템에 있어서,

상기 이니시에이터는 상기 기억 영역에 대한 판독 및 기록 액세스에 대응하는 명령을 상기 타겟에 의해 보유될 수 있는 판독 및 기록 명령의 수를 초과하지 않도록 상기 타겟으로 전송하고,

상기 타겟은 상기 수신된 판독 및 기록 명령들을 상이한 큐에 보유하고, 상기 보유된 명령들을 독립적으로 처리하는 것을 특징으로 하는 통신 시스템.

청구항 10.

제9항에 있어서, 상기 타겟은 1의 용량을 갖는 큐에 상기 기록 명령을 보유하고 상기 큐 내에 있는 상기 명령을 처리하는 것을 특징으로 하는 통신 시스템.

청구항 11.

제9항에 있어서, 상기 기억 영역 내에서의 데이타 기록 요청을 상기 타겟으로부터 수신시, 상기 이니시에이터는 상기 요청을 제2 큐 내에 보유하고, 상기 타겟에 기록 명령을 발행하는 것을 특징으로 하는 통신 시스템.

청구항 12.

제11항에 있어서, 상기 타겟은 상기 제2 큐에 의해 보유될 수 있는 상기 요청의 수를 초과하지 않도록 상기 이니시에이터에 기록 요청을 발행하는 것을 특징으로 하는 통신 시스템.

청구항 13.

제9항에 있어서, 상기 이니시에이터가 상기 타겟으로부터의 요청에 응답하여 상기 기억 영역 내에 기록 명령을 발행하는 것을 특징으로 하는 통신 시스템.

청구항 14.

제13항에 있어서, 상기 이니시에이터에 전송될 데이타가 선정된 크기보다 작은 경우, 상기 타겟은 상기 기억 영역 내에 기록할 것을 상기 이니시에이터에 대하여 요청하는 것이 아니라, 상기 이니시에이터 내에 있는 상기 기억 영역의 선정된 영역 내에 상기 데이타를 직접 기록하는 것을 특징으로 하는 통신 시스템.

청구항 15.

이니시에이터의 기억 영역 액세스시 타겟으로부터 데이타를 교환하기 위한 통신 시스템에 있어서,

상기 타겟은 전송될 데이타의 크기가 선정된 크기를 초과하는지 여부를 체크하여, 상기 전송될 데이타의 크기가 상기 선정된 크기를 초과하는 경우 상기 기억 영역 내에 기록 명령을 발행하도록 상기 이니시에이터에 대하여 요청하고, 전송될 상기 데이타의 크기가 선정된 크기를 초과하지 않는 경우 상기 데이타를 상기 이니시에이터에 송신하며,

상기 이니시에이터는 상기 타겟으로부터 기록 명령 발행 요청을 수신하면 기록 명령을 발행하는 것을 특징으로 하는 통신 시스템.

청구항 16.

제9항에 있어서, 상기 타겟 및 이니시에이터는 채널 식별자를 상기 기록 및 판독 명령에 부여하고, 채널 식별자 유닛들마다 상기 명령들을 독립적으로 처리하는 것을 특징으로 하는 통신 시스템.

청구항 17.

통신에 의해 접속된 타겟으로부터 메모리 내의 기억 영역 액세스시 상기 타겟과 데이타를 교환하는 통신 제어 방법에 있어서,

상기 타겟으로부터의 자발적인 요청을 수신하여 큐 내에 상기 요청을 큐잉(queing)하는 큐잉 단계,

상기 타겟에 의해 보유될 수 있는 판독 및 기록 명령의 수를 초과하지 않도록 애플리케이션 또는 상기 타겟으로부터의 요청에 응답하여 상기 기억 영역에 상기 판독 및 기록 명령을 생성하여 전송하는 명령 생성 단계

를 포함하는 것을 특징으로 하는 통신 제어 방법.

청구항 18.

통신에 의해 접속된 이니시에이터의 기억 영역 액세스시 상기 이니시에이터와 데이타를 교환하는 통신 제어 방법에 있어서,

선정된 크기를 갖는 큐 내에 상기 이니시에이터로부터 수신된 판독 명령을 큐잉하는 큐잉 단계,

상기 큐로부터의 판독 명령을 픽업(picking up)하여 실행하는 순차 실행(queued execution) 단계,

상기 이니시에이터로부터 수신된 기록 명령을 수신 후 즉시 실행하는 즉시 실행 단계, 및

데이타 전송 요청을 상기 이니시에이터에 발행하는 전송 요청 단계

를 포함하는 것을 특징으로 하는 통신 제어 방법.

청구항 19.

제17항에 있어서, 채널 식별자를 사용하여 상기 타겟으로부터의 상기 요청에 대응하는 채널을 판별(discriminating)하는 채널 판별 단계를 더 포함하고, 상기 명령 생성 단계는 상기 판별된 채널 또는 애플리케이션에 대응하는 채널에 대응하는 채널 식별자를 상기 명령에 부여하는 단계를 포함하는 것을 특징으로 하는 통신 제어 방법.

청구항 20.

제18항에 있어서, 채널 식별자를 사용하여 상기 이니시에이터로부터의 상기 명령에 대응하는 채널을 판별하는 채널 판별 단계, 및 상기 판별된 채널에 대응하는 채널 식별자가 부여된 응답을 상기 이니시에이터에 역송신하는 응답 단계를 더 포함하되, 상기 전송 요청 단계는 상기 판별된 채널 또는 애플리케이션에 대응하는 채널에 대응하는 채널 식별자를 상기 요청에 부여하는 단계를 더 포함하는 것을 특징으로 하는 통신 제어 방법.

청구항 21.

제20항에 있어서, 상기 전송 요청 단계는 상기 이니시에이터에 대한 데이타 크기가 선정된 크기보다 더 클 경우 상기 기억 영역 내에 데이타 기록 명령을 발행할 것을 상기 이니시에이터에 대하여 요청하고, 상기 데이타의 크기가 상기 선정된 크기보다 더 작을 경우에는 상기 기억 영역의 선정된 영역 내에 상기 데이타를 직접 기록하는 단계를 포함하는 것을 특징으로 하는 통신 제어 방법.

청구항 22.

기억 영역을 통하여 타겟과 데이타를 교환하기 위한 통신 제어 장치에 있어서,

타겟과 통신하기 위한 수단,

상기 기억 영역을 포함하는 메모리,

상기 타겟으로부터 자발적인 요청을 큐잉하기 위한 큐 관리 수단, 및

애플리케이션 또는 상기 타겟으로부터의 요청에 응답하여, 상기 타겟에 의해 보유될 수 있는 판독 및 기록 명령의 수를 초과하지 않도록 상기 기억 영역에 대한 판독 및 기록 명령을 생성하여 전송하는 명령 생성 수단

을 포함하는 것을 특징으로 하는 통신 제어 장치.

청구항 23.

이니시에이터의 기억 영역을 액세스함으로써 상기 이니시에이터와 데이타를 교환하기 위한 통신 제어 장치에 있어서,

상기 이니시에이터와 통신하기 위한 수단,

상기 이니시에이터로부터 수신된 판독 명령을 보유하는 선정된 크기의 큐,

상기 큐로부터 상기 판독 명령을 픽업하여 실행하는 순차 실행(queued execution) 수단,

상기 이니시에이터로부터 수신된 기록 명령을 수신 후 즉시 실행하기 위한 즉시 실행 수단, 및

데이타 전송 요청을 상기 이니시에이터에 발행하기 위한 전송 요청 수단

을 포함하는 것을 특징으로 하는 통신 제어 장치.

청구항 24.

제22항에 있어서, 채널 식별자를 사용하여 상기 타겟으로부터의 상기 요청에 대응하는 채널을 판별(discriminating)하는 채널 판별 수단을 더 포함하고,

상기 명령 생성 수단은 상기 판별된 채널 또는 애플리케이션에 대응하는 채널에 대응하는 채널 식별자를 상기 명령에 부여하는 것을 특징으로 하는 통신 제어 장치.

청구항 25.

제23항에 있어서,채널 식별자를 사용하여 상기 이니시에이터로부터의 상기 명령에 대응하는 채널을 판별하는 채널 판별 수단, 및

상기 판별된 채널에 대응하는 채널 식별자가 부여된 응답을 상기 이니시에이터에 역송신하는 응답 수단을 더 포함하되,

상기 전송 요청 수단은 상기 판별된 채널 또는 애플리케이션에 대응하는 채널에 대응하는 채널 식별자를 상기 요청에 부여하는 것을 특징으로 하는 통신 제어 장치.

청구항 26.

제25항에 있어서, 상기 전송 요청 수단은 상기 이니시에이터에 대한 데이타 크기가 선정된 크기보다 더 클 경우 상기 기억 영역 내에 데이타 기록 명령을 발행할 것을 상기 이니시에이터에 대하여 요청하고, 상기 데이타의 크기가 상기 선정된 크기보다 더 작을 경우 상기 기억 영역의 선정된 영역 내에 상기 데이타를 직접 기록하는 것을 특징으로 하는 통신 제어 장치.

청구항 27.

통신에 의해 타겟으로부터 기억 영역을 액세스함으로써 데이타를 교환하기위한 통신 제어 프로그램을 기억하는 컴퓨터 판독가능 기억 매체에 있어서, 상기 프로그램은

상기 타겟으로부터의 자발적인 요청을 큐잉하기 위한 큐 관리 처리 단계, 및

애플리케이션 또는 상기 타겟으로부터의 요청에 응답하여, 상기 타겟에 의해 보유될 수 있는 판독 및 기록 명령의 수를 초과하지 않도록 상기 기억 영역에 대한 판독 및 기록 명령을 생성하여 전송하기 위한 명령 생성 처리 단계

를 포함하는 것을 특징으로 하는 컴퓨터 판독가능 기억 매체.

청구항 28.

통신에 의해 접속된 이니시에이터의 기억 영역을 액세스함으로써 데이타를 교환하기 위한 통신 제어 프로그램을 기억하는 컴퓨터 판독가능 기억 매체에 있어서, 상기 프로그램은

상기 이니시에이터로부터 수신된 판독 명령을 선정된 용량을 갖는 큐 내에 큐잉하기 위한 큐 관리 처리 단계,

상기 큐로부터 상기 판독 명령을 픽업하여 실행하는 순차 실행 처리 단계,

상기 이니시에이터로부터 수신된 기록 명령을 수신 후 즉시 실행하기 위한 즉시 실행 처리 단계, 및

데이타 전송 요청을 상기 이니시에이터에 발행하기 위한 전송 요청 처리 단계

를 포함하는 것을 특징으로 하는 컴퓨터 판독가능 기억 매체.

청구항 29.

제27항에 있어서, 상기 프로그램은 채널 식별자를 사용하여 상기 타겟으로부터의 상기 요청에 대응하는 채널을 판별하는 채널 판별 처리 단계를 더 포함하고,

상기 명령 생성 처리 단계는 상기 판별된 채널 또는 애플리케이션에 대응하는 채널에 대응하는 채널 식별자를 상기 명령에 부여하는 것을 특징으로 하는 컴퓨터 판독가능 기억 매체.

청구항 30.

제28항에 있어서, 상기 프로그램은 채널 식별자를 사용하여 상기 이니시에이터로부터의 상기 명령에 대응하는 채널을 판별하는 채널 판별 처리 단계, 및

상기 판별된 채널에 대응하는 채널 식별자가 부여된 응답을 상기 이니시에이터에 역송신하는 응답 처리 단계를 더 포함하고,

상기 전송 요청 처리 단계는 상기 판별된 채널 또는 애플리케이션에 대응하는 채널에 대응하는 채널 식별자를 상기 요청에 부여하는

것을 특징으로 하는 컴퓨터 판독가능 기억 매체.

청구항 31.

제30항에 있어서, 상기 전송 요청 처리 단계는 상기 이니시에이터에 대한 데이타 크기가 선정된 크기보다 더 클 경우 상기 기억 영역 내에 데이타 기록 명령을 발행할 것을 상기 이니시에이터에 대하여 요청하고, 상기 데이타의 크기가 상기 선정된 크기보다 더 작을 경우 상기 기억 영역의 선정된 영역 내에 상기 데이타를 직접 기록하는 것을 특징으로 하는 컴퓨터 판독가능 기억 매체.

청구항 32.

제1항의 통신 제어 방법을 사용하는 인쇄 시스템에 있어서, 이니시에이터로서 역할하는 호스트 장치는 타겟으로서 역할하는 프린터 장치에 접속되고, 상기 프린터 장치는 상기 호스트 장치로부터 인쇄 데이타를 수신하고 상기 수신된 인쇄 데이타에 기초하여 인쇄 출력하며, 상기 호스트 장치는 상기 프린터 장치의 상태 정보(status information)를 수신하는 것을 특징으로 하는 인쇄 시스템.

청구항 33.

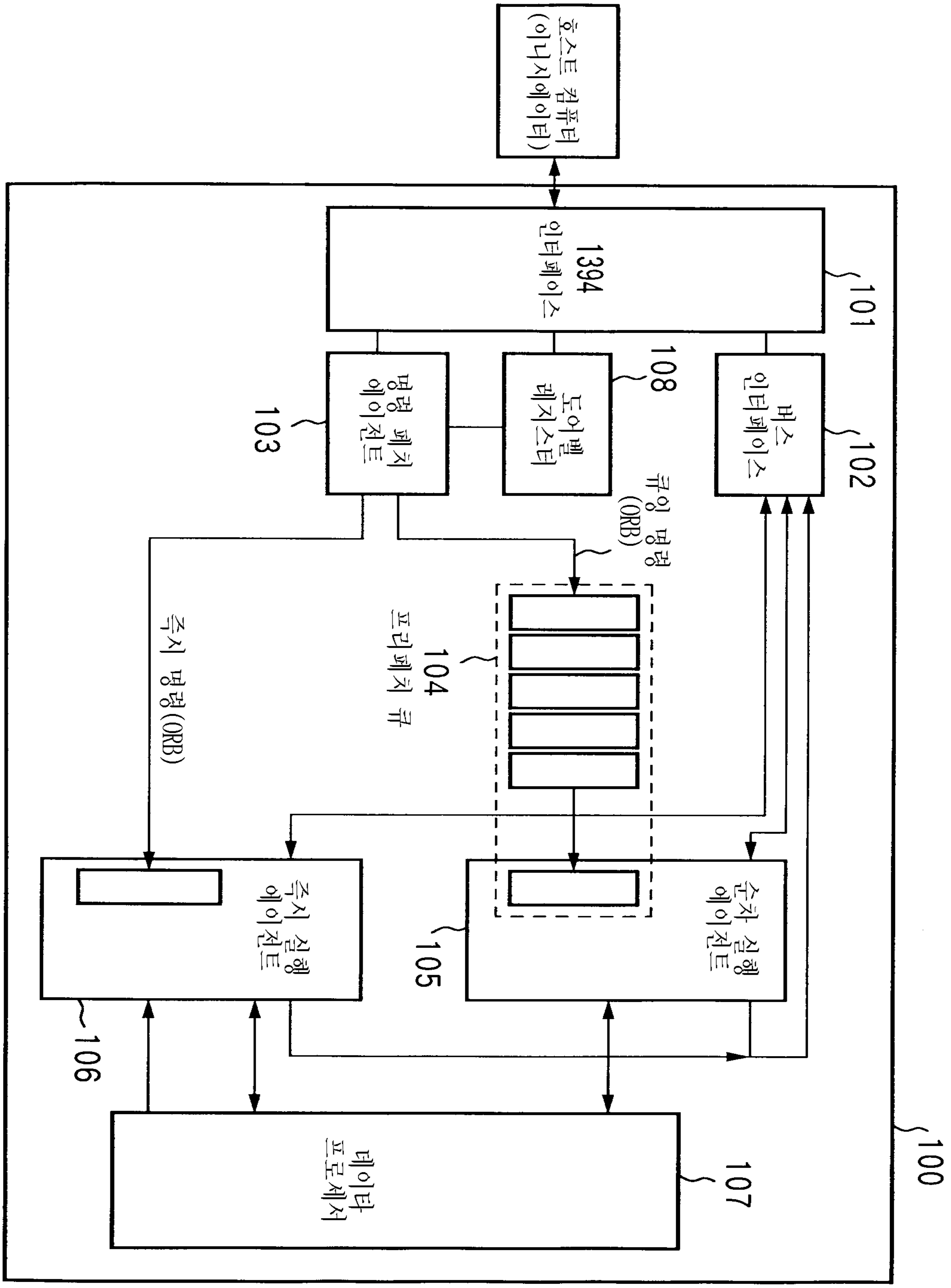
제17항의 통신 제어 방법에 의해 인쇄 데이타를 타겟에 전송하고, 상기 타겟으로부터 상태 정보를 수신하기 위한 인쇄 제어 장치.

청구항 34.

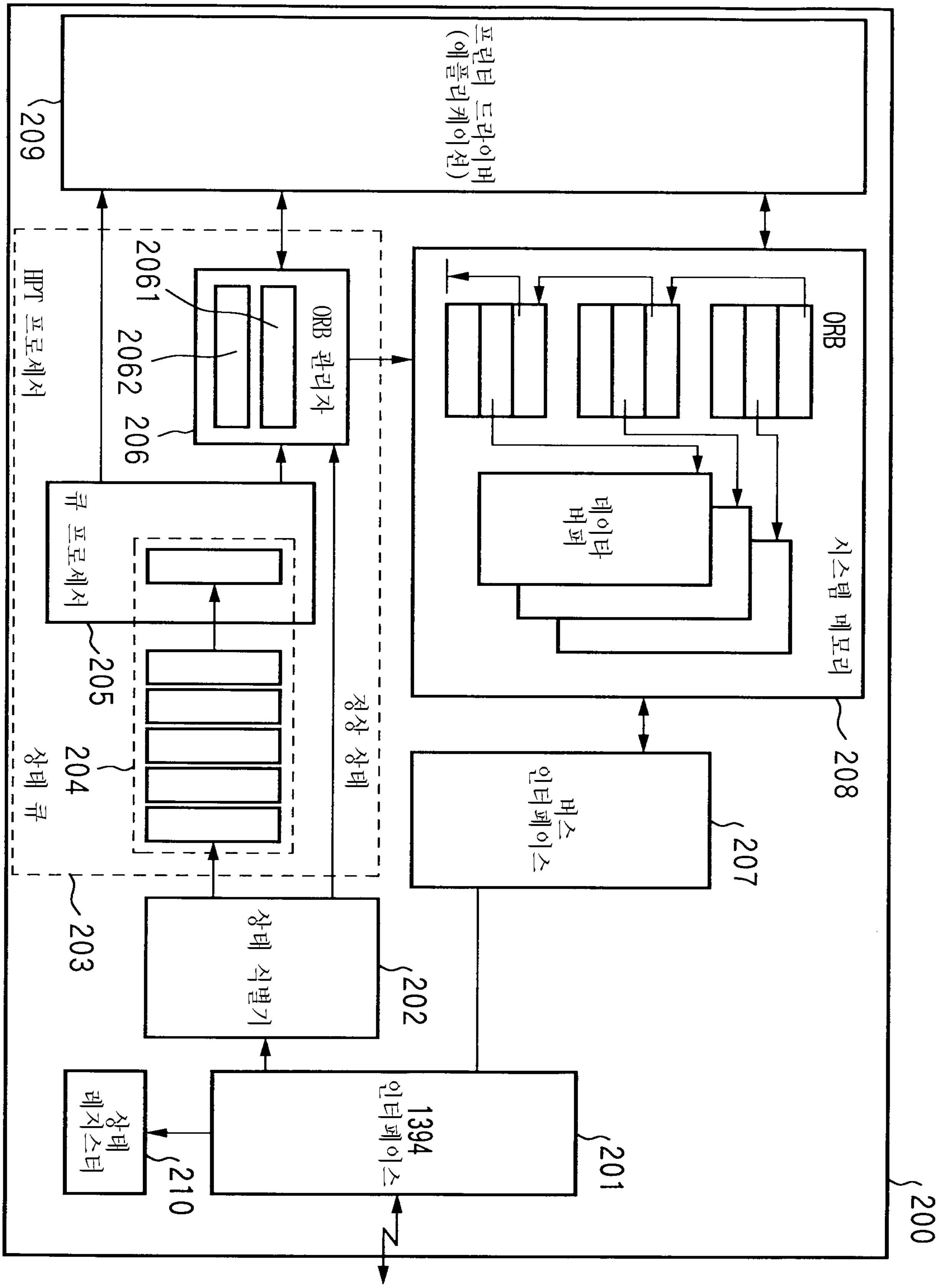
제18항의 통신 제어 방법에 의해 이니시에이터로부터 인쇄 데이타를 수신하고, 상태 정보를 상기 이니시에이터로 전송하기 위한 인쇄 장치.

도면

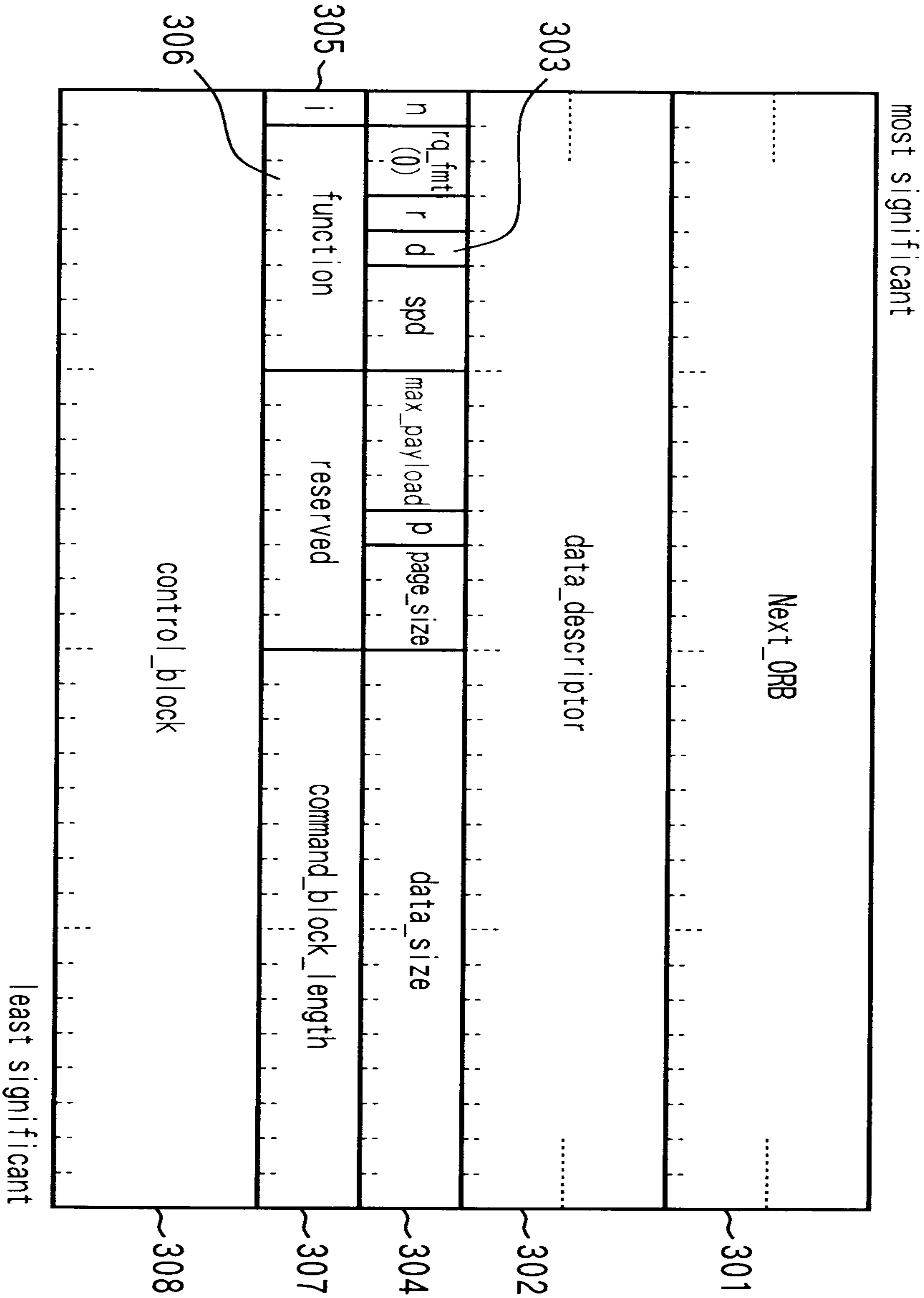
도면 1


호스트 컴퓨터
(이니시에이터)
1394
인터페이스
101
버스
인터페이스
102
명령 페치
에이전트
103
도어벨
레지스터
108
큐잉 명령
(ORB)
104
프리페치 큐
즉시 명령(ORB)
순차 실행
에이전트
105
즉시 실행
에이전트
106
데이터
프로세서
107
100

도면 2


프린터 드라이버
(애플리케이션)
209
ORB 관리자
2061
2062
206
HPT 프로세서
큐 프로세서
205
204
상태 큐
203
정상 상태
ORB
시스템 메모리
데이터
버퍼
208
버스
인터페이스
207
상태 식별기
202
1394
인터페이스
201
상태
레지스터
210
200

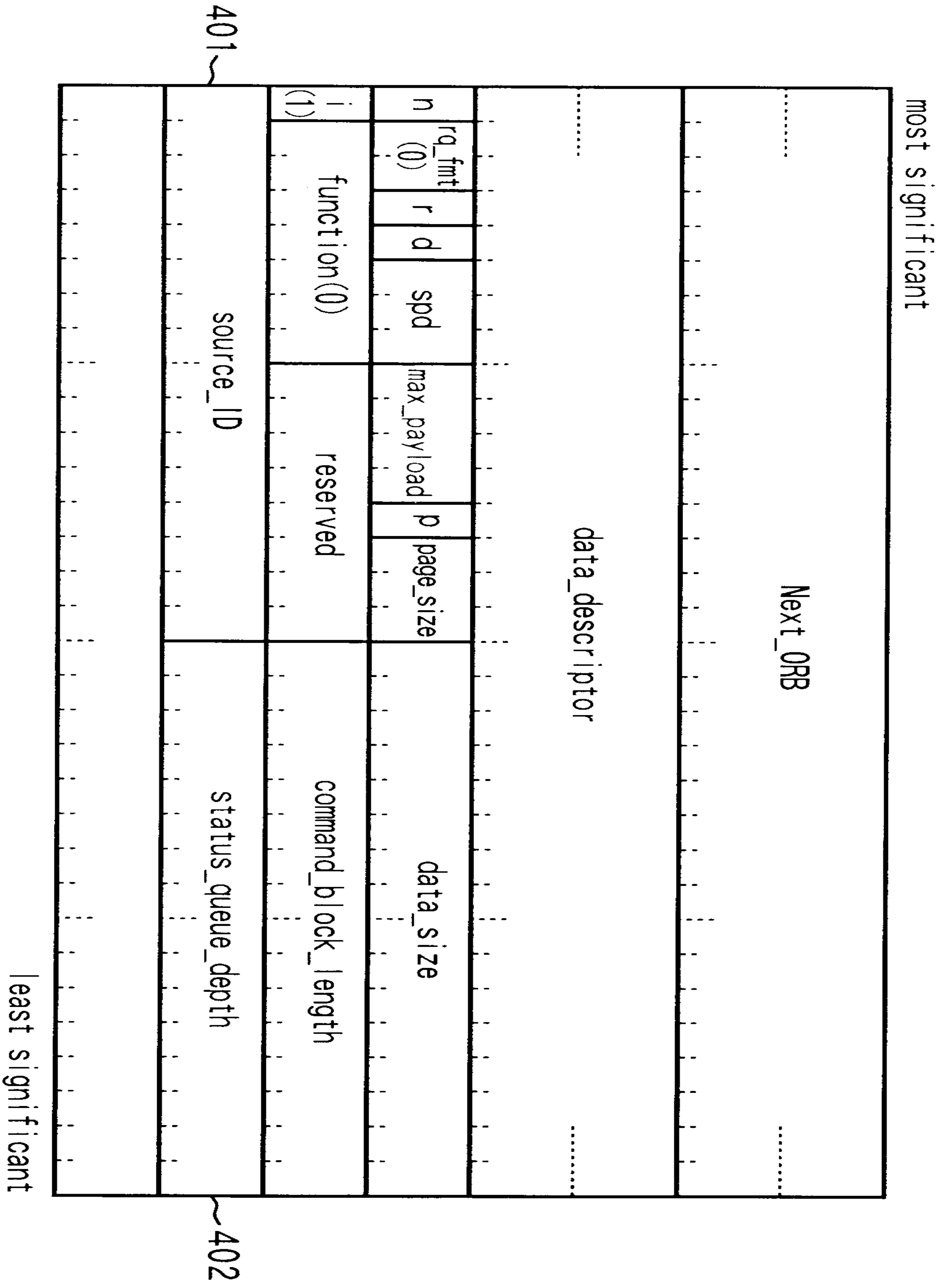
도면 3a


most significant
Next_ORB
data_descriptor
n
rq_fmt (0)
r
d
spd
max_payload
p
page_size
data_size
i
function
reserved
command_block_length
control_block
least significant
301
302
303
304
305
306
307
308

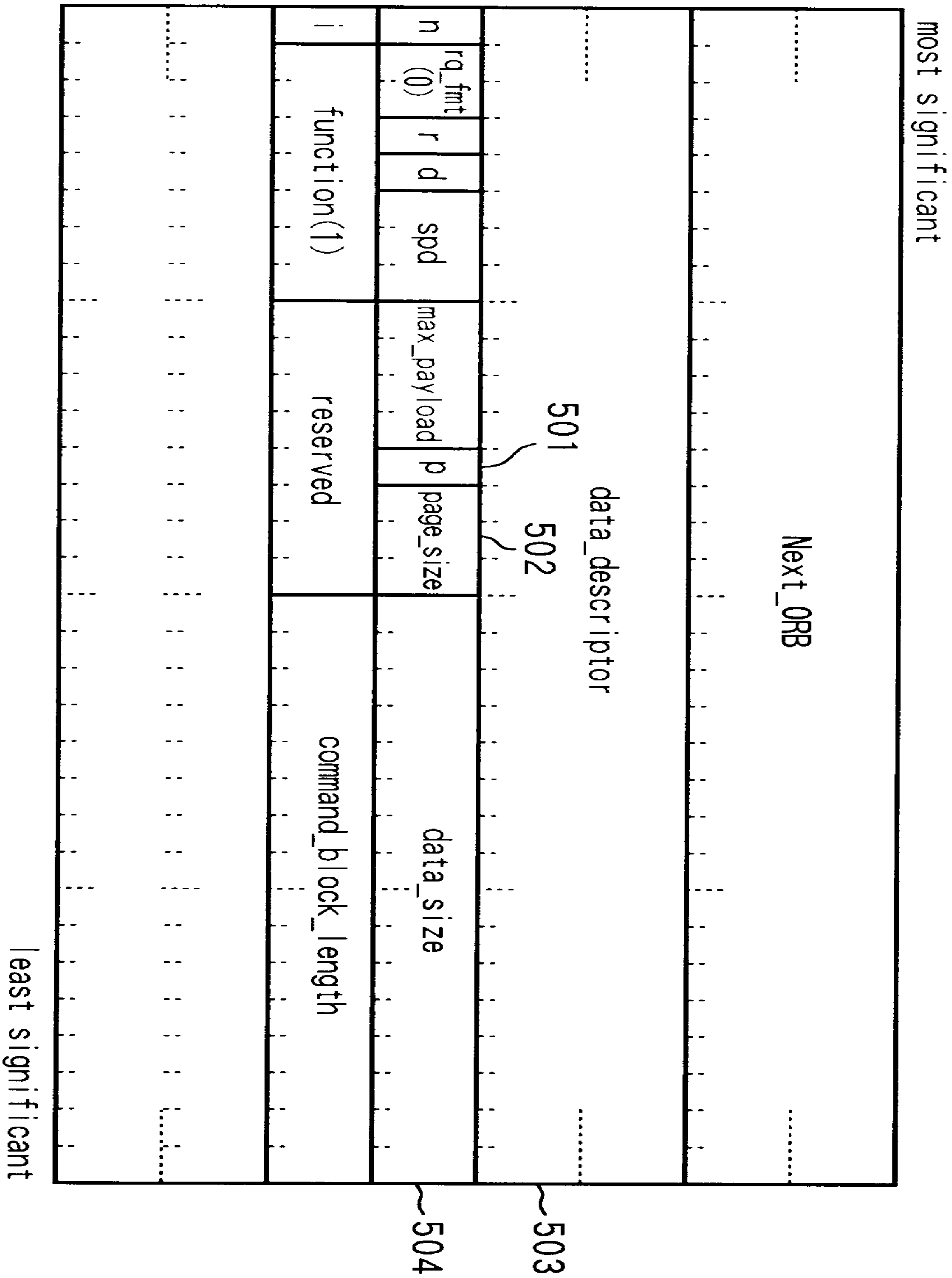
도면 3b

값	HPT 명령
0	큐 깊이
1	데이타 전송
2	판독 요청
3	직접 상태 응답
4-7	예비
8	디바이스 리소스 획득
9	디바이스 리소스 해제
A16	기본 디바이스 상태
B_{16}-F_{16}	예비
10_{16}-$7F_{16}$	제어 세트에 의존

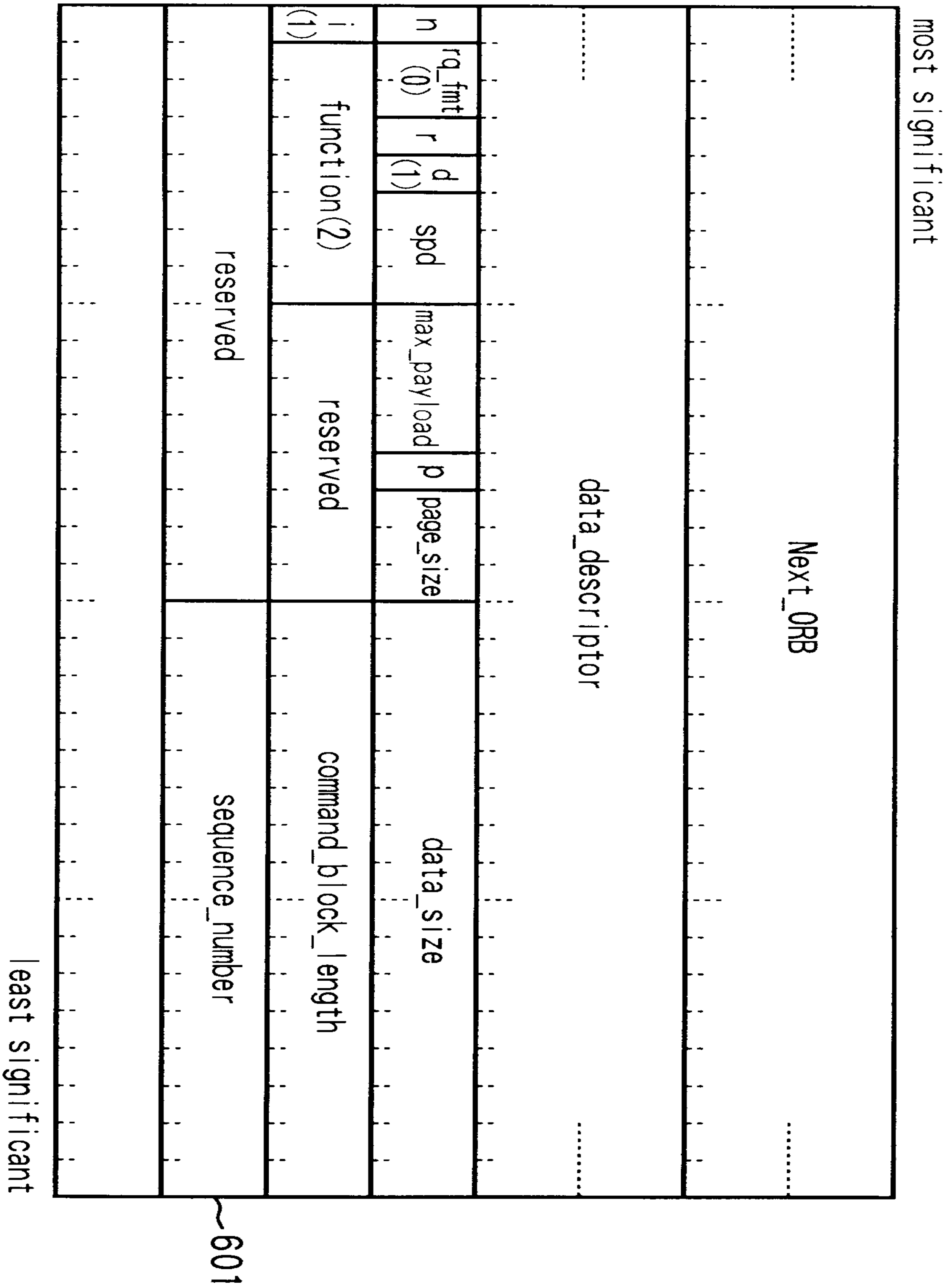
도면 4


most significant
Next_ORB
data_descriptor
n
rq_fmt (0)
r
d
spd
max_payload
p
page_size
data_size
i (1)
function(0)
reserved
command_block_length
source_ID
status_queue_depth
least significant
401
402

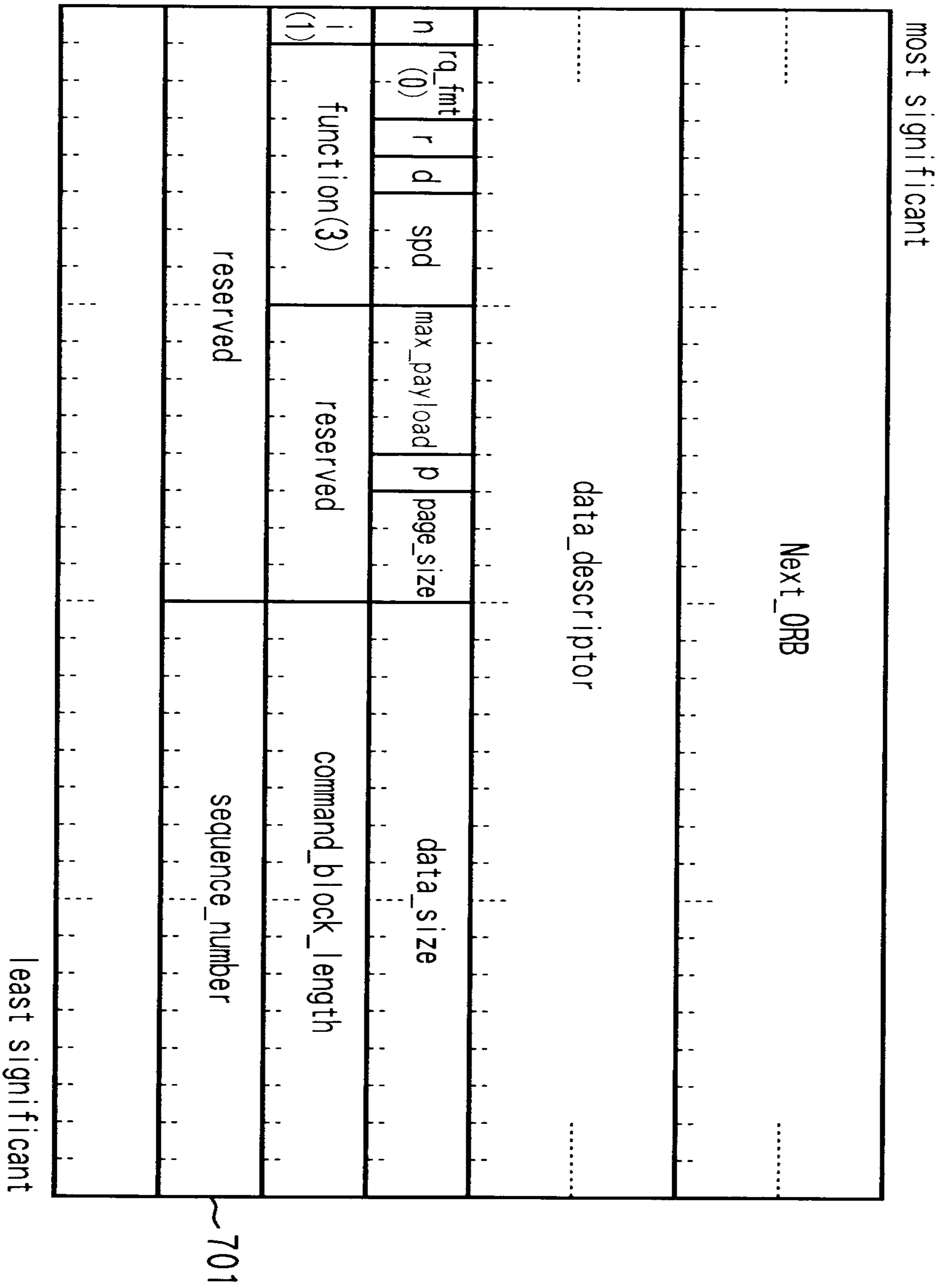
도면 5


most significant
Next_ORB
data_descriptor
n
rq_fmt (0)
r
d
spd
max_payload
p
page_size
data_size
501
502
503
504
l
function(1)
reserved
command_block_length
least significant

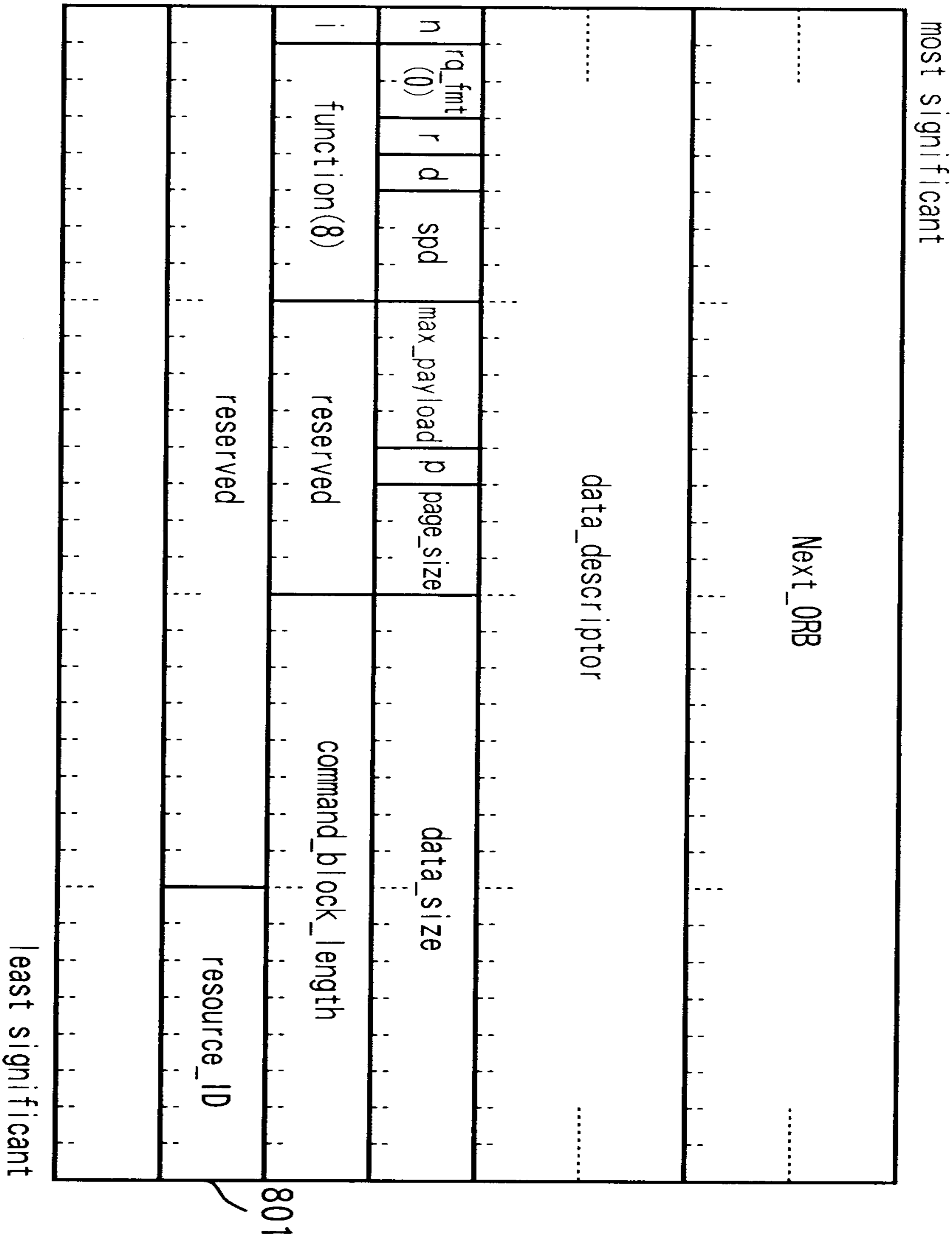
도면 6


most significant
Next_ORB
data_descriptor
n
rq_fmt (0)
r
d (1)
spd
max_payload
p
page_size
data_size
i (1)
function(2)
reserved
command_block_length
reserved
sequence_number
least significant
601

도면 7


most significant
Next_ORB
data_descriptor
n
rq_fmt (0)
r
d
spd
max_payload
p
page_size
data_size
i (1)
function(3)
reserved
command_block_length
reserved
sequence_number
least significant
701

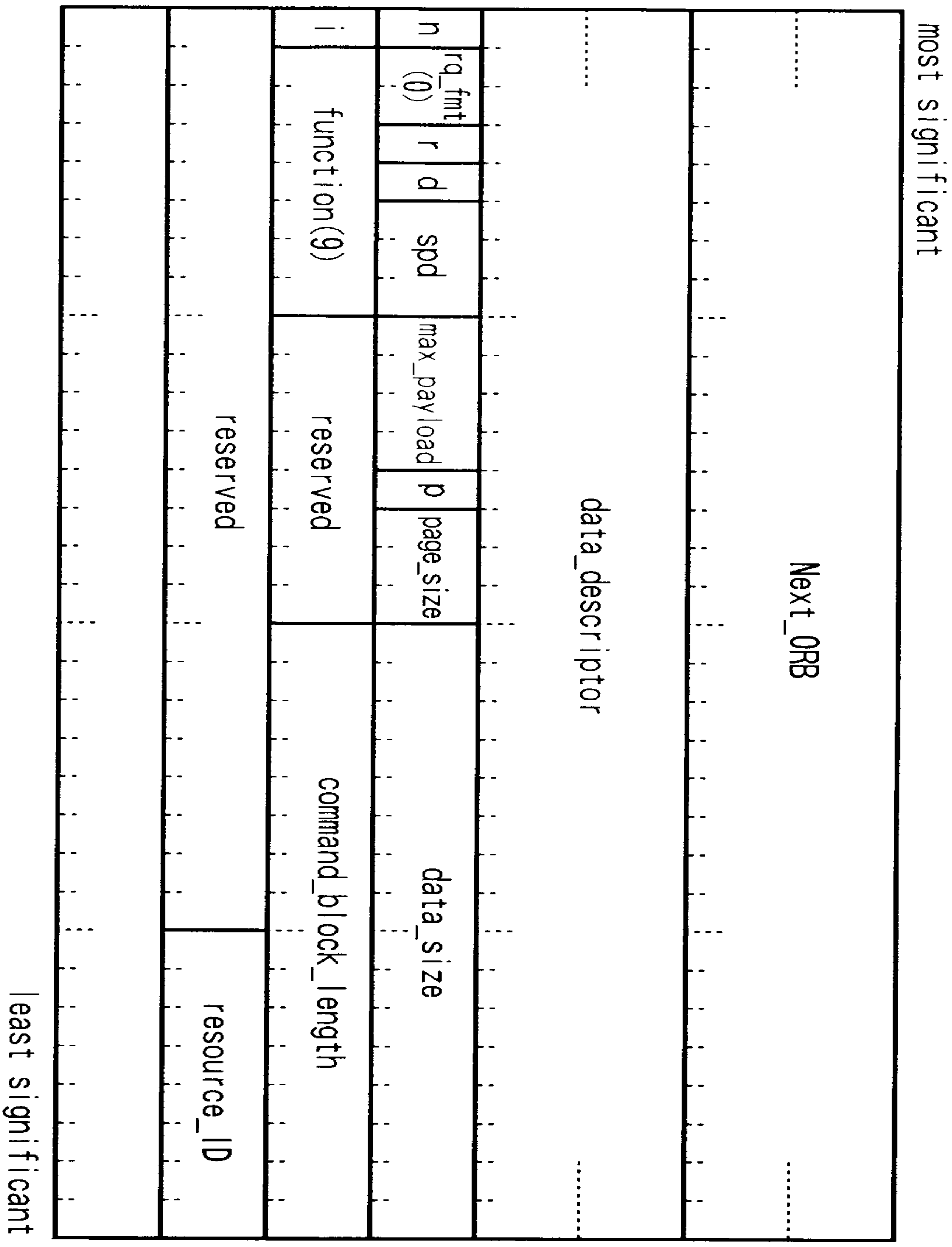
도면 8a


most significant
Next_ORB
data_descriptor
n
rq_fmt (0)
r
d
spd
max_payload
p
page_size
data_size
i
function(8)
reserved
command_block_length
reserved
resource_ID
least significant
801

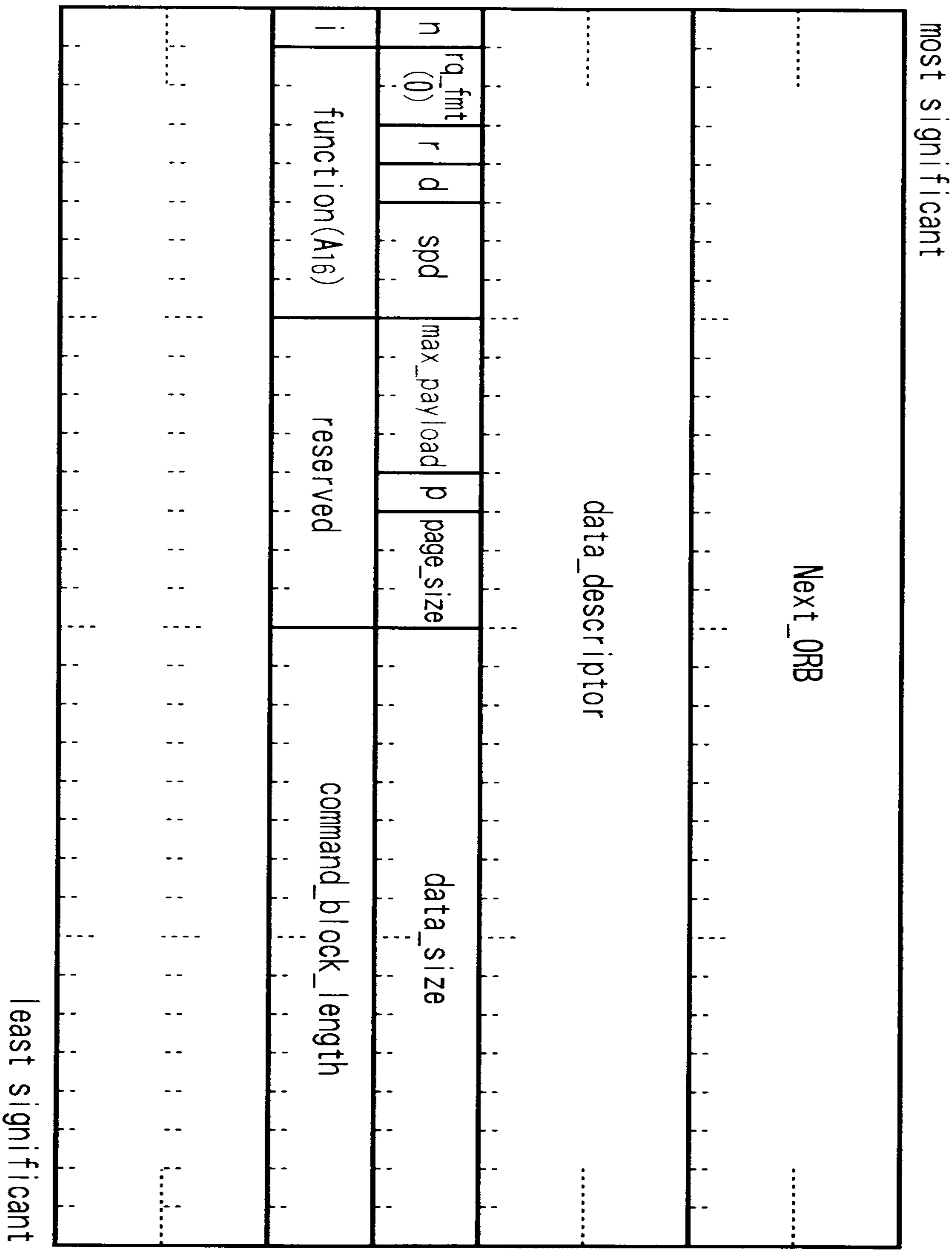
도면 8b

값	리소스
0	디바이스 클래스 및 논리 유닛 특성에 의존
$1\text{-}F_{16}$	예비
$10_{16}\text{-}7_{16}$	제어 세트에 의존

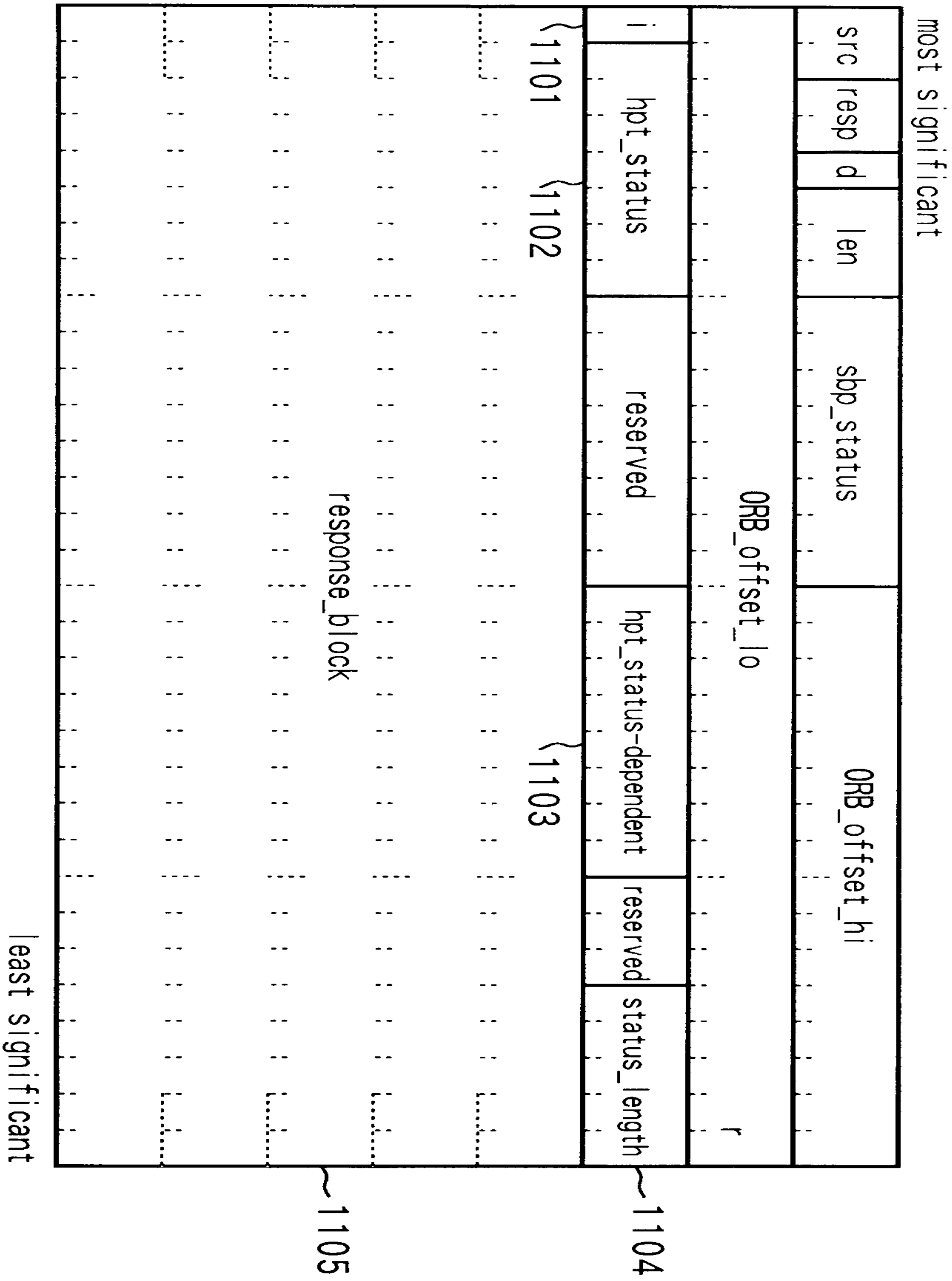
도면 9


most significant
Next_ORB
data_descriptor
n
rq_fmt (0)
r
d
spd
max_payload
p
page_size
data_size
i
function(9)
reserved
command_block_length
reserved
resource_ID
least significant

도면 10


most significant
Next_ORB
data_descriptor
n
rq_fmt (0)
r
d
spd
max_payload
p
page_size
data_size
i
function(A16)
reserved
command_block_length
least significant

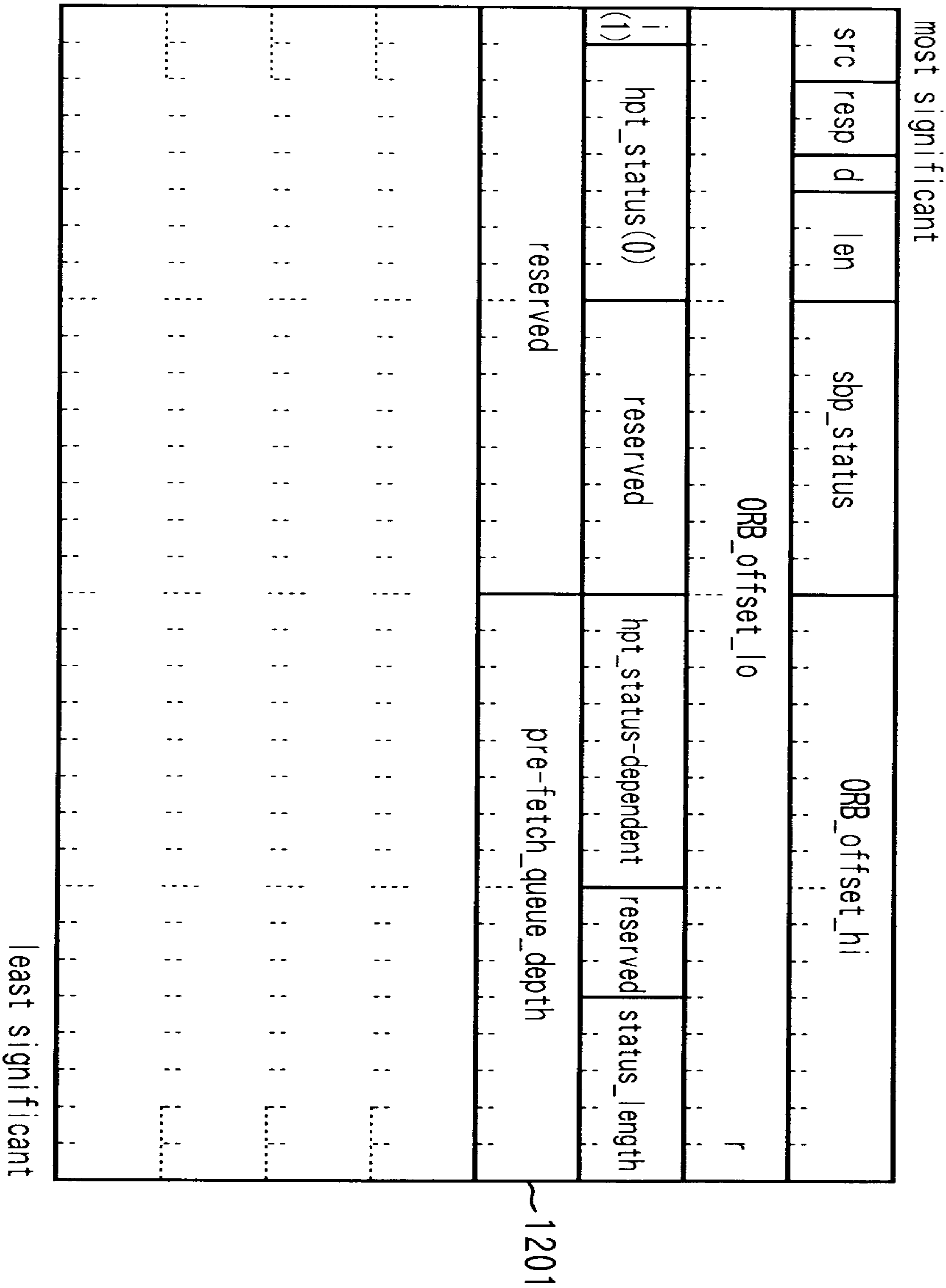
도면 11a


most significant
src
resp
d
len
sbp_status
ORB_offset_hi
ORB_offset_lo
r
i
hpt_status
reserved
hpt_status-dependent
reserved
status_length
response_block
least significant
1101
1102
1103
1104
1105

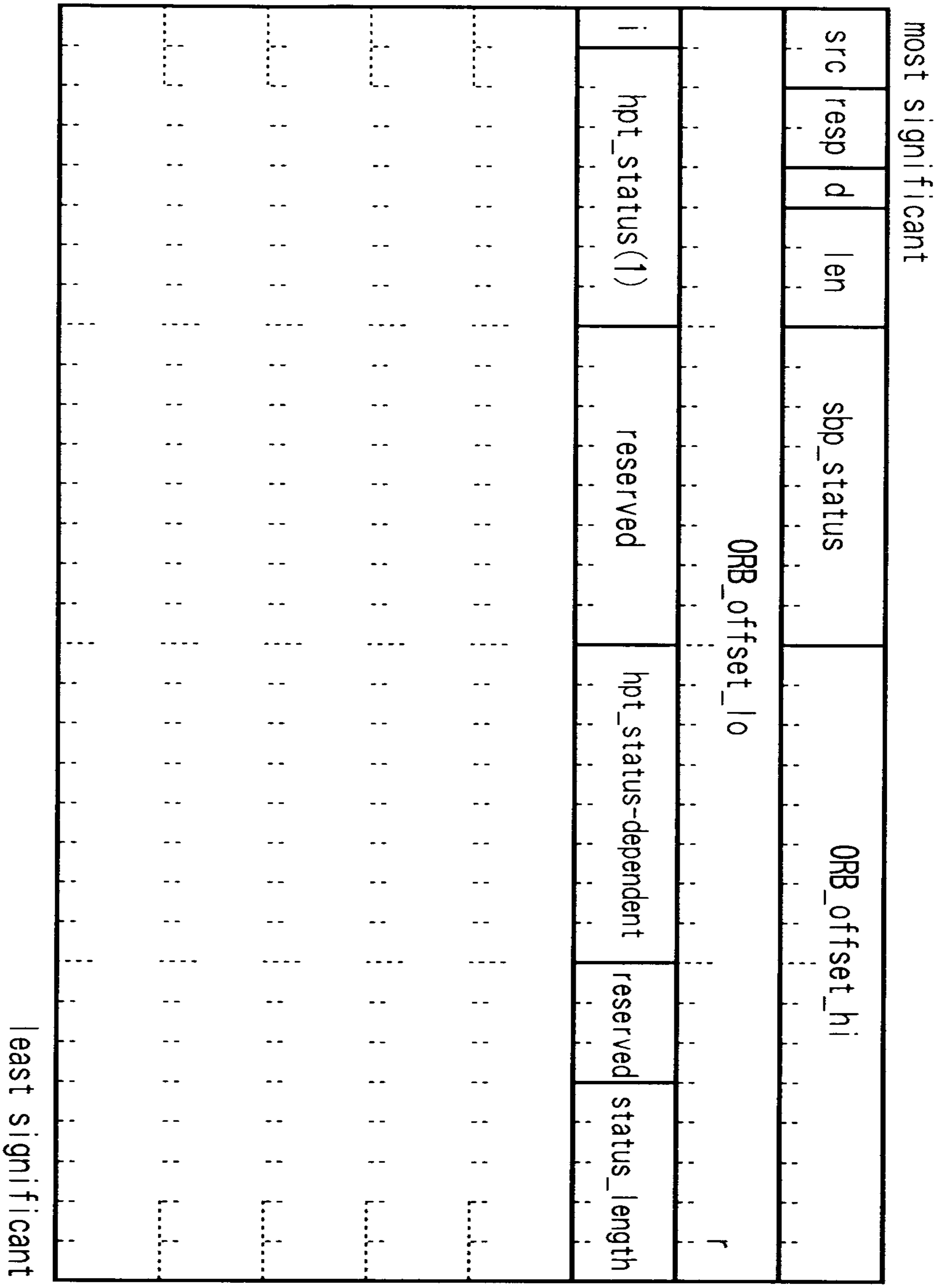
도면 11b

값	HPT 상태
0	큐 깊이
1	데이타 전송
2	판독 요청
3	직접
4-7	예비
8	디바이스 리소스
9	디바이스 리소스
A16	기본 디바이스
B_{16}-F_{16}	예비
10_{16}-$7F_{16}$	제어 세트에 의존

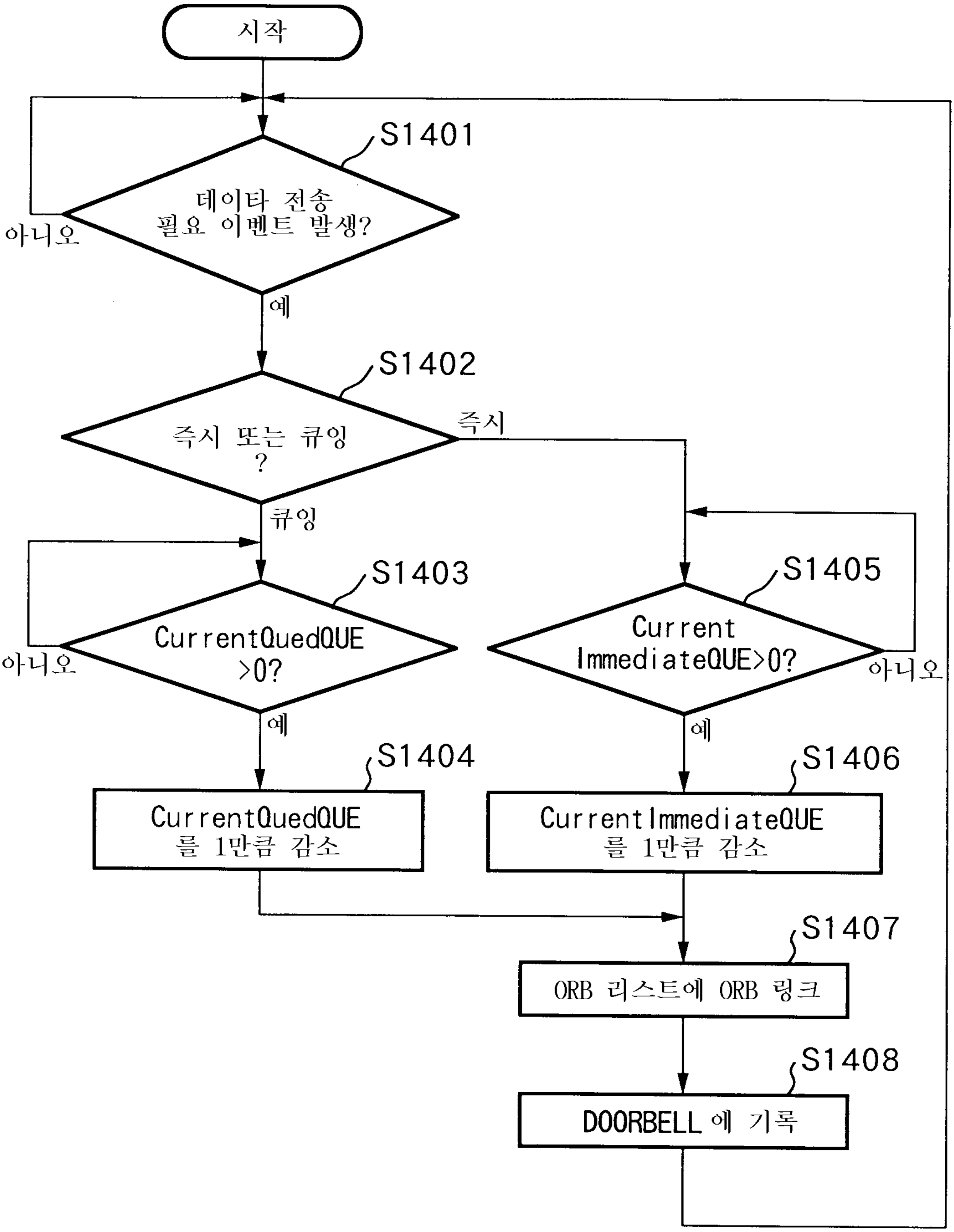
도면 12


most significant
src
resp
d
len
sbp_status
ORB_offset_hi
ORB_offset_lo
r
i
(1)
hpt_status(0)
reserved
hpt_status-dependent
reserved
status_length
reserved
pre-fetch_queue_depth
least significant
1201

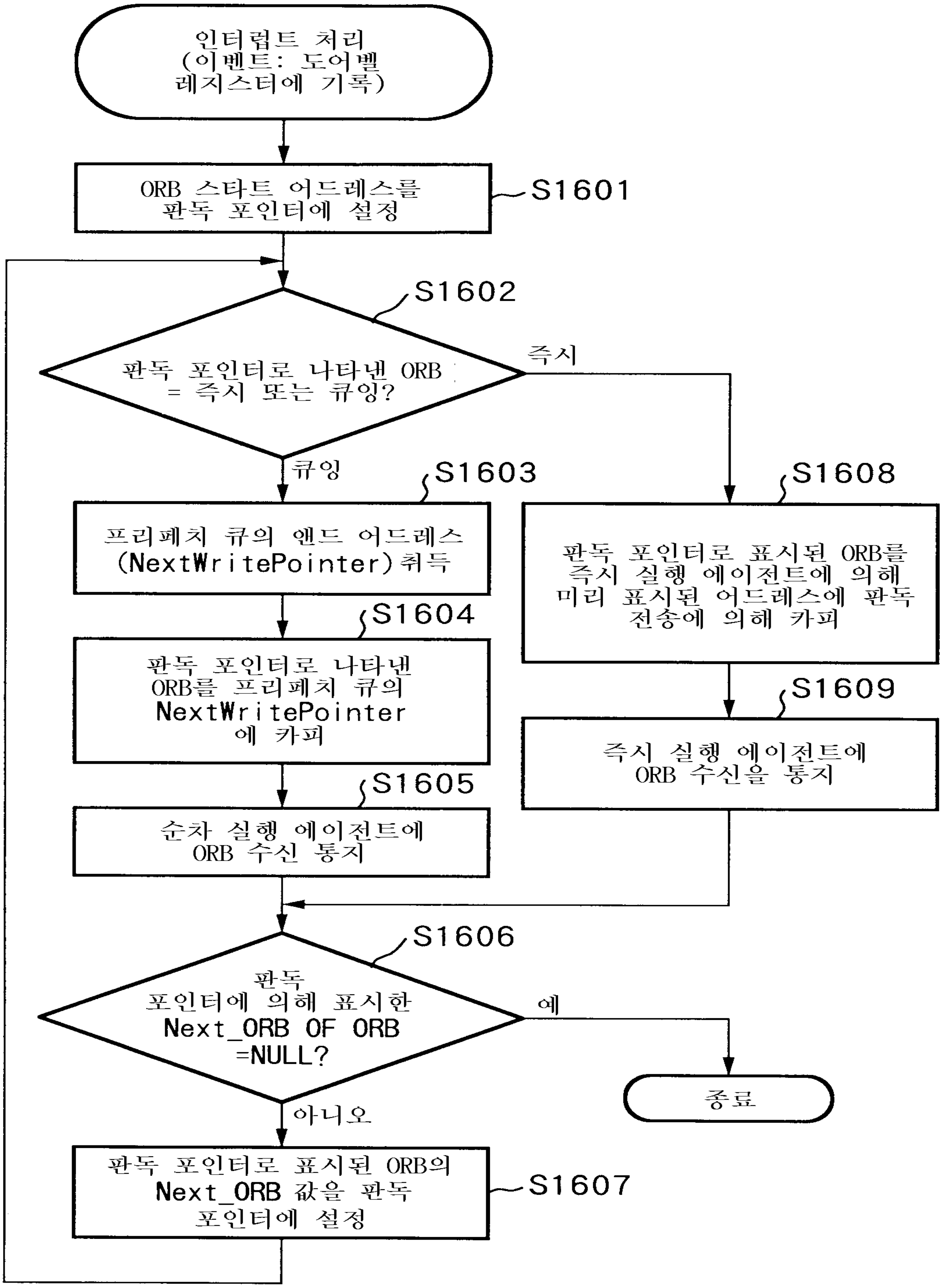
도면 13


most significant
src
resp
d
len
sbp_status
ORB_offset_hi
ORB_offset_lo
r
i
hpt_status(1)
reserved
hpt_status-dependent
reserved
status_length
least significant

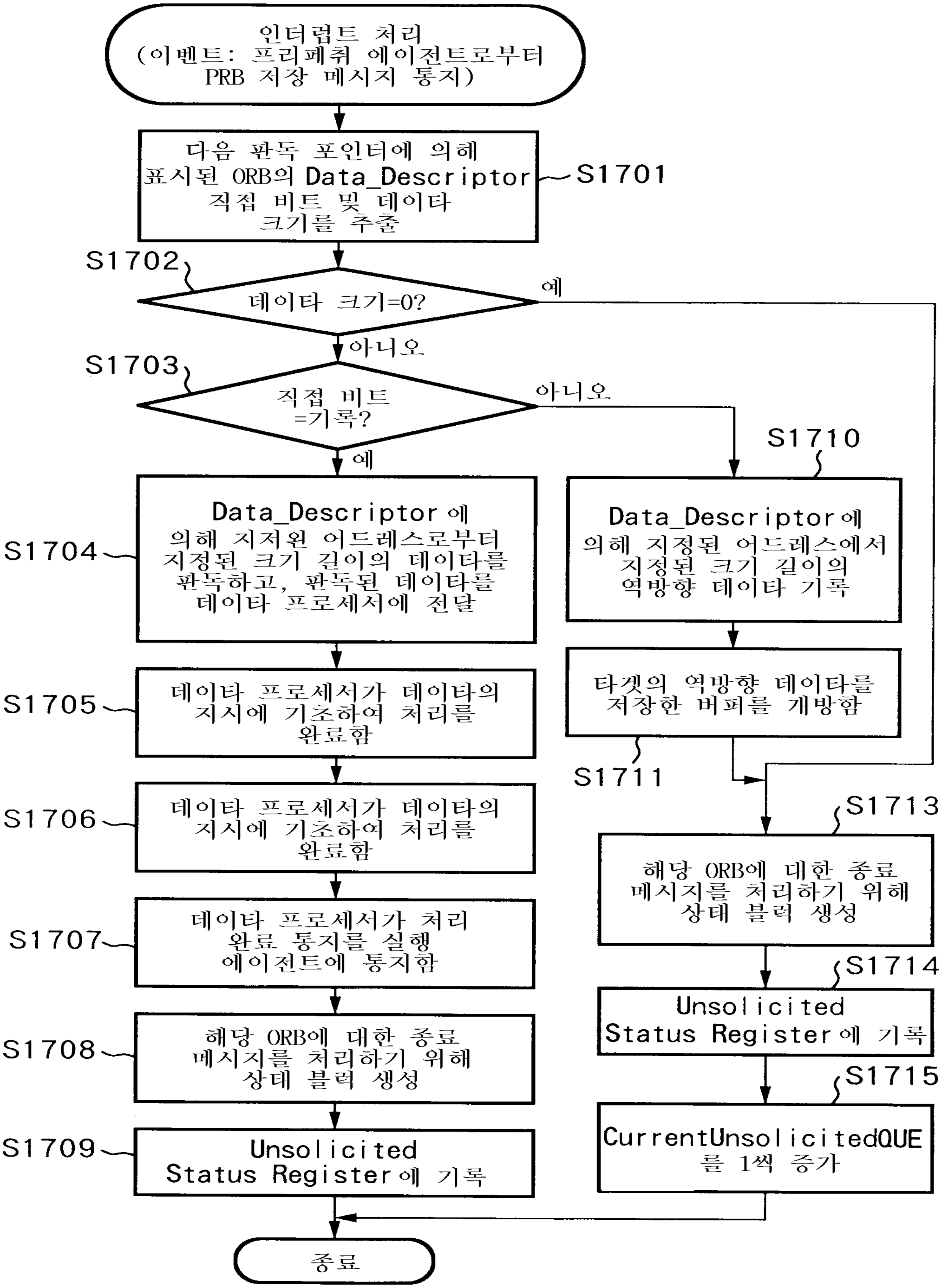
도면 14


시작
S1401
데이타 전송 필요 이벤트 발생?
아니오
예
S1402
즉시 또는 큐잉 ?
즉시
큐잉
S1403
CurrentQuedQUE >0?
아니오
예
S1405
Current ImmediateQUE>0?
아니오
예
S1404
CurrentQuedQUE 를 1만큼 감소
S1406
CurrentImmediateQUE 를 1만큼 감소
S1407
ORB 리스트에 ORB 링크
S1408
DOORBELL 에 기록

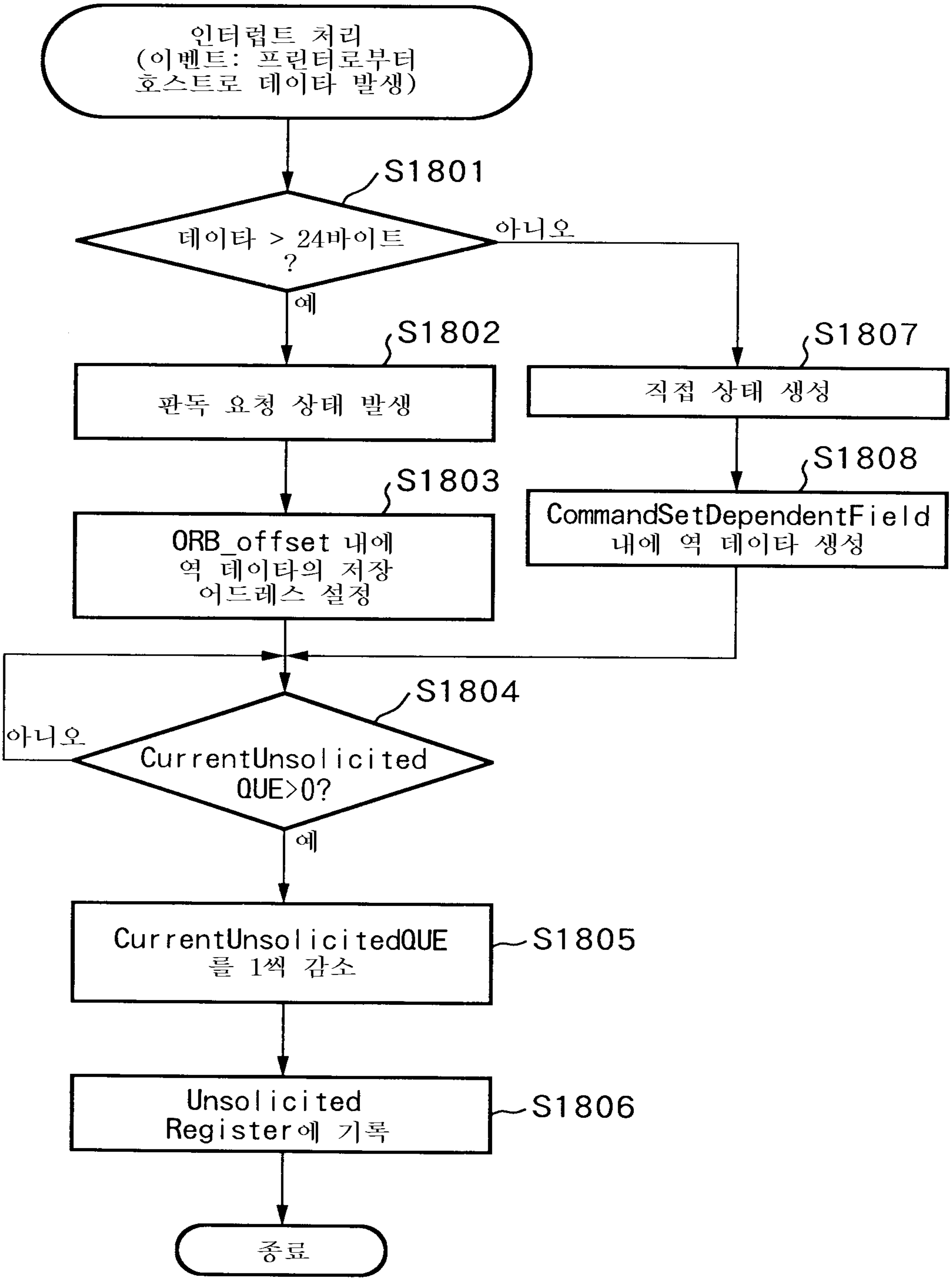
도면 15


인터럽트 처리 (이벤트: 도어벨 레지스터에 기록)
ORB 스타트 어드레스를 판독 포인터에 설정
S1601
S1602
판독 포인터로 나타낸 ORB = 즉시 또는 큐잉?
즉시
큐잉
S1603
프리페치 큐의 앤드 어드레스 (NextWritePointer) 취득
S1604
판독 포인터로 나타낸 ORB를 프리페치 큐의 NextWritePointer 에 카피
S1605
순차 실행 에이전트에 ORB 수신 통지
S1608
판독 포인터로 표시된 ORB를 즉시 실행 에이전트에 의해 미리 표시된 어드레스에 판독 전송에 의해 카피
S1609
즉시 실행 에이전트에 ORB 수신을 통지
S1606
판독 포인터에 의해 표시한 Next_ORB OF ORB =NULL?
예
종료
아니오
판독 포인터로 표시된 ORB의 Next_ORB 값을 판독 포인터에 설정
S1607

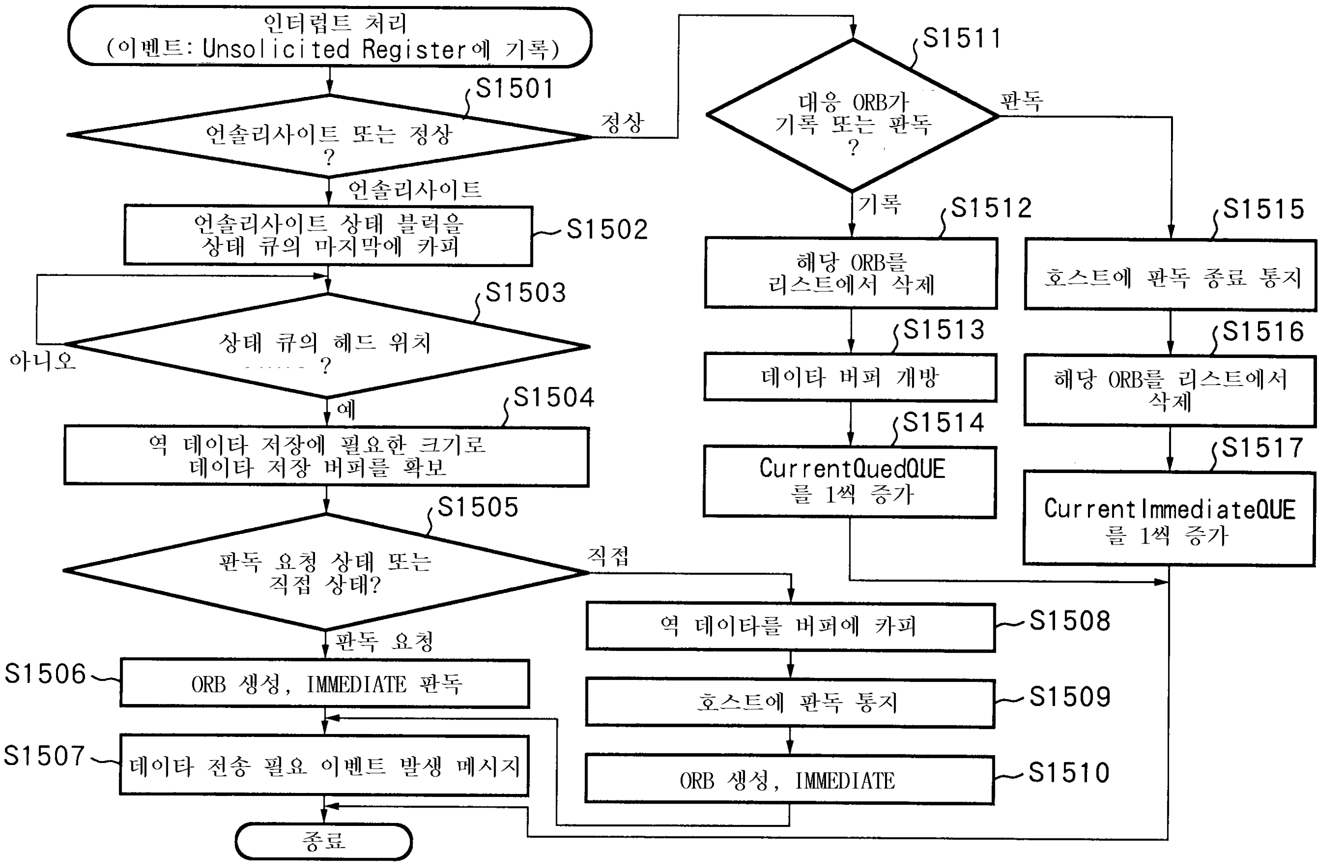
도면 16


인터럽트 처리
(이벤트: 프리페취 에이전트로부터
PRB 저장 메시지 통지)
다음 판독 포인터에 의해
표시된 ORB의 Data_Descriptor
직접 비트 및 데이타
크기를 추출
S1701
S1702
데이타 크기=0?
예
아니오
S1703
직접 비트
=기록?
아니오
예
S1710
S1704
Data_Descriptor에
의해 지저된 어드레스로부터
지정된 크기 길이의 데이타를
판독하고, 판독된 데이타를
데이타 프로세서에 전달
Data_Descriptor에
의해 지정된 어드레스에서
지정된 크기 길이의
역방향 데이타 기록
S1705
데이타 프로세서가 데이타의
지시에 기초하여 처리를
완료함
타겟의 역방향 데이타를
저장한 버퍼를 개방함
S1711
S1713
S1706
데이타 프로세서가 데이타의
지시에 기초하여 처리를
완료함
해당 ORB에 대한 종료
메시지를 처리하기 위해
상태 블럭 생성
S1707
데이타 프로세서가 처리
완료 통지를 실행
에이전트에 통지함
S1714
Unsolicited
Status Register에 기록
S1708
해당 ORB에 대한 종료
메시지를 처리하기 위해
상태 블럭 생성
S1715
CurrentUnsolicitedQUE
를 1씩 증가
S1709
Unsolicited
Status Register에 기록
종료

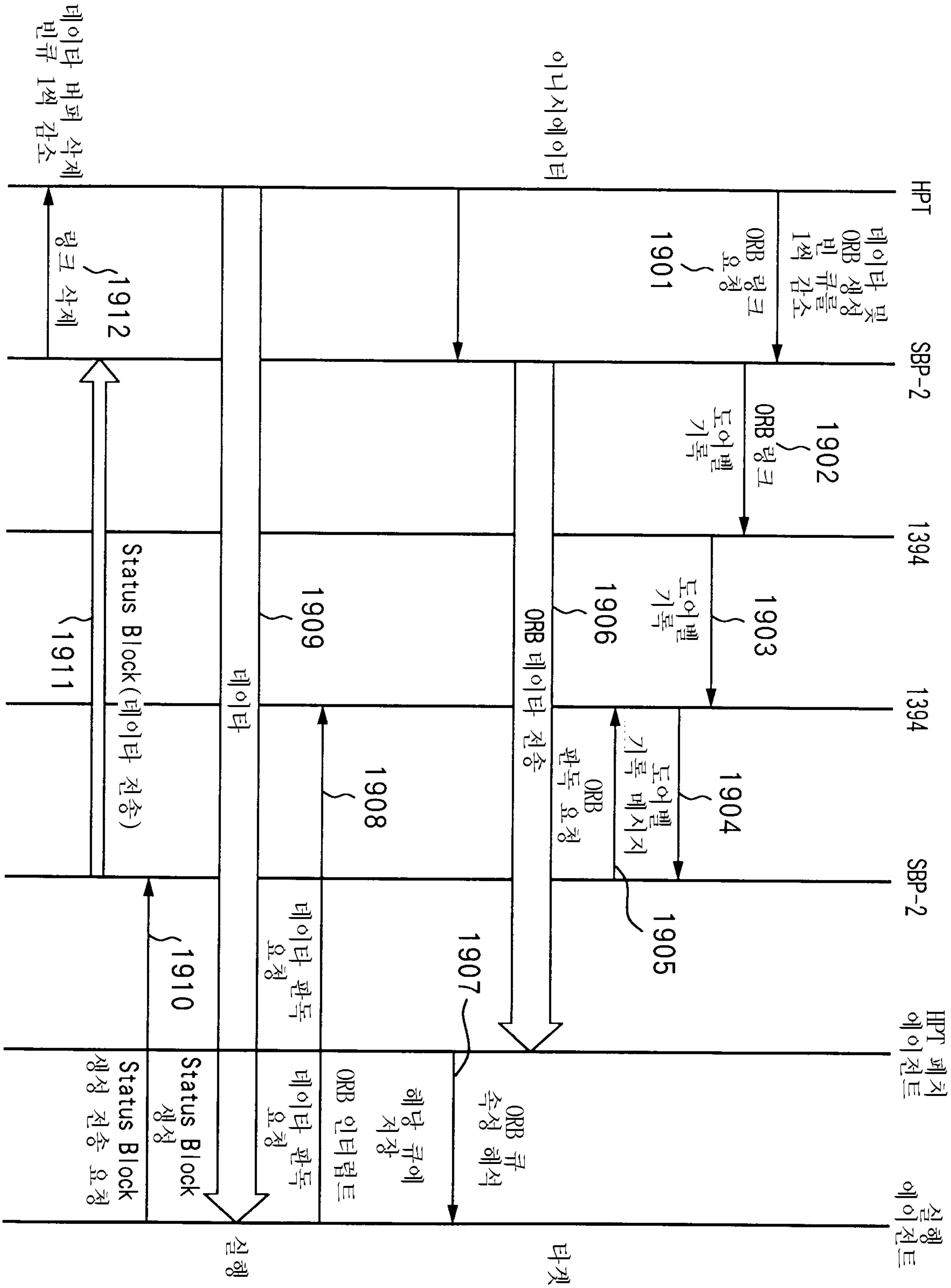
도면 17


인터럽트 처리
(이벤트: 프린터로부터
호스트로 데이타 발생)
S1801
데이타 > 24바이트
?
아니오
예
S1802
판독 요청 상태 발생
S1807
직접 상태 생성
S1803
ORB_offset 내에
역 데이타의 저장
어드레스 설정
S1808
CommandSetDependentField
내에 역 데이타 생성
S1804
아니오
CurrentUnsolicited
QUE>0?
예
CurrentUnsolicitedQUE
를 1씩 감소
S1805
Unsolicited
Register에 기록
S1806
종료

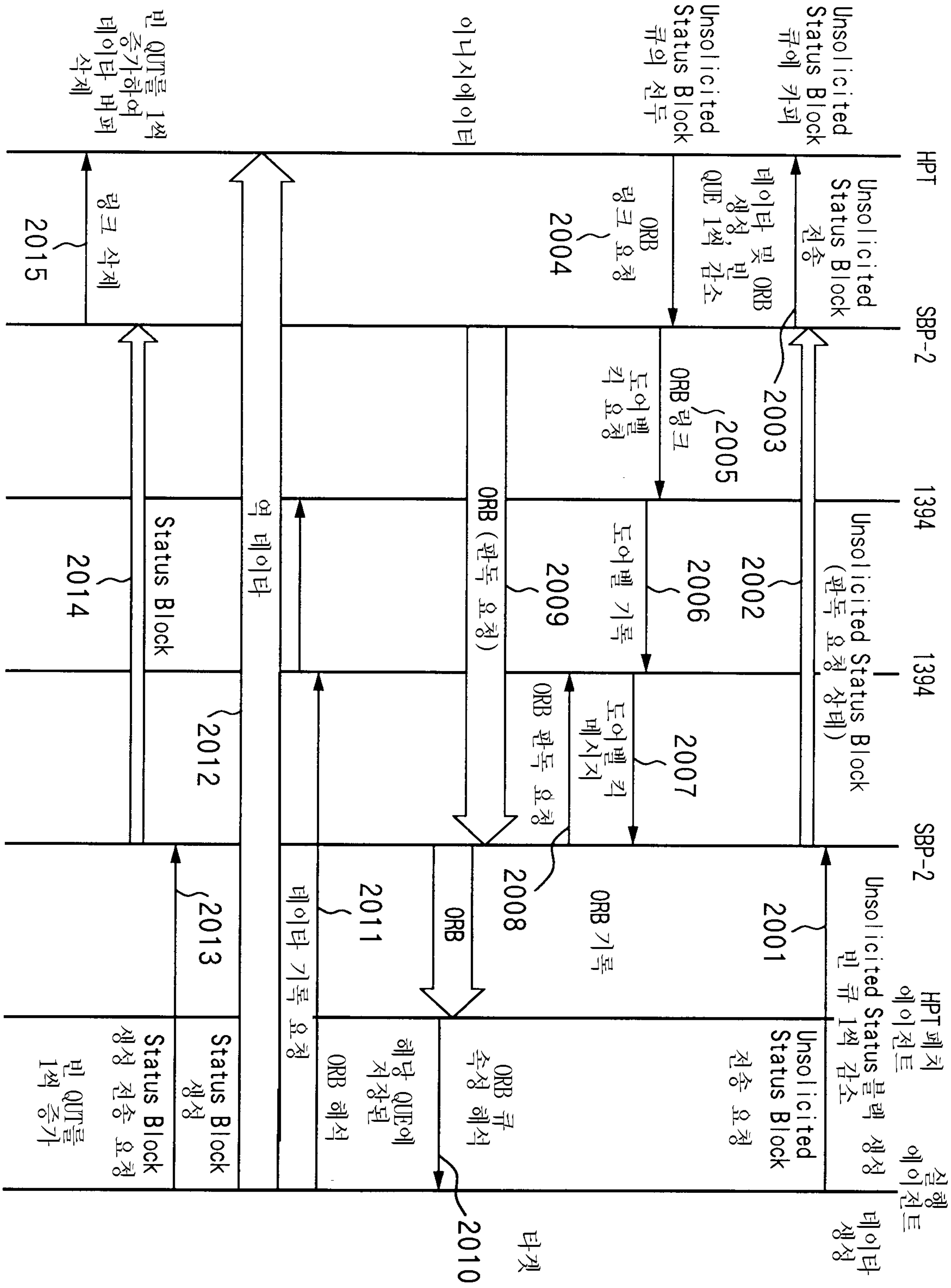
도면 18


인터럽트 처리
(이벤트: Unsolicited Register에 기록)
S1501
언솔리사이트 또는 정상 ?
정상
언솔리사이트
언솔리사이트 상태 블럭을 상태 큐의 마지막에 카피
S1502
S1503
상태 큐의 헤드 위치 ?
아니오
예
S1504
역 데이타 저장에 필요한 크기로 데이타 저장 버퍼를 확보
S1505
판독 요청 상태 또는 직접 상태?
직접
판독 요청
S1506
ORB 생성, IMMEDIATE 판독
S1507
데이타 전송 필요 이벤트 발생 메시지
종료
S1511
대응 ORB가 기록 또는 판독 ?
판독
기록
S1512
해당 ORB를 리스트에서 삭제
S1513
데이타 버퍼 개방
S1514
CurrentQuedQUE를 1씩 증가
S1515
호스트에 판독 종료 통지
S1516
해당 ORB를 리스트에서 삭제
S1517
CurrentImmediateQUE를 1씩 증가
역 데이타를 버퍼에 카피
S1508
호스트에 판독 통지
S1509
ORB 생성, IMMEDIATE
S1510

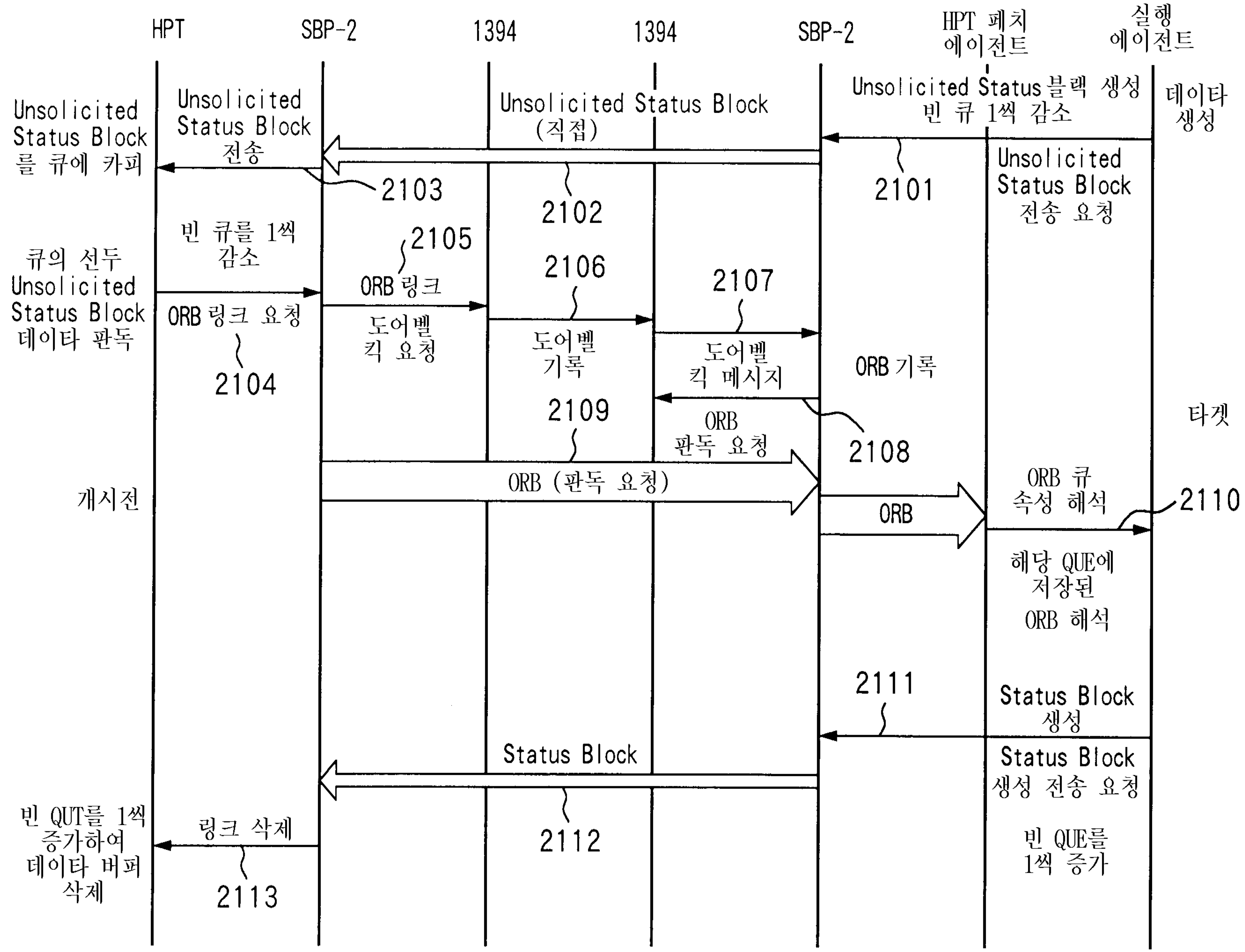
도면 19


HPT
SBP-2
1394
1394
SBP-2
HPT 페치 에이전트
실행 에이전트
이니시에이터
타겟
실행
데이타 및 ORB 생성 빈 큐블 1씩 감소
1901
ORB 링크 요청
1902
ORB 링크 도어벨 기록
1903
도어벨 기록
1904
도어벨 기록 메시지
1905
ORB 판독 요청
1906
ORB 데이타 전송
1907
ORB 큐 수신 해석
해당 큐에 저장
ORB 인터럽트
1908
데이타 판독 요청
1909
데이타
Status Block 생성
1910
Status Block 생성 전송 요청
1911
Status Block(데이타 전송)
1912
링크 삭제
데이타 버퍼 삭제 빈큐 1씩 감소

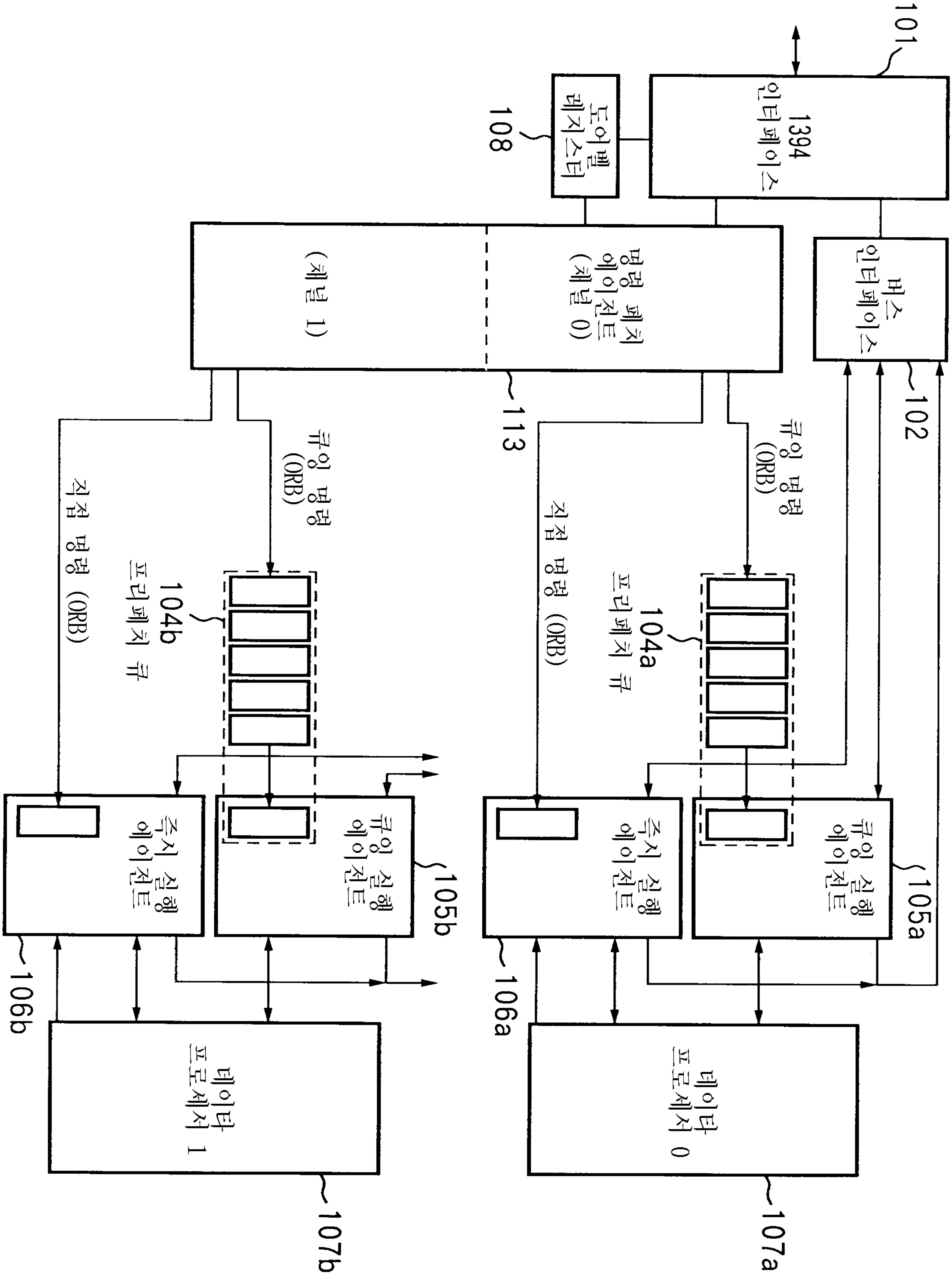
도면 20


HPT
SBP-2
1394
1394
SBP-2
HPT패키지 에이전트
실행 에이전트
타깃
Unsolicited Status Block 큐에 가짐
Unsolicited Status Block 큐의 선두
이니시에이터
빈 QUT를 1씩 증가하여 데이터 버퍼 삭제
데이터 생성
Unsolicited Status블럭 생성 빈 큐 1씩 감소
Unsolicited Status Block 전송 요청
ORB 큐 수신 해석
해당 QUE에 저장된 ORB 해석
Status Block 생성
Status Block 생성 전송 요청
빈 QUT를 1씩 증가
2001
2002
Unsolicited Status Block (판독 요청 상태)
2003
Unsolicited Status Block 전송
데이터 및 ORB 생성, 빈 QUE 1씩 감소
2004
ORB 링크 요청
2005
ORB 링크
도어벨 기록 요청
2006
도어벨 기록
2007
도어벨 기록 메시지
ORB 판독 요청
2008
ORB 기록
2009
ORB (판독 요청)
2010
ORB
2011
데이터 기록 요청
2012
역 데이터
2013
2014
Status Block
2015
링크 삭제

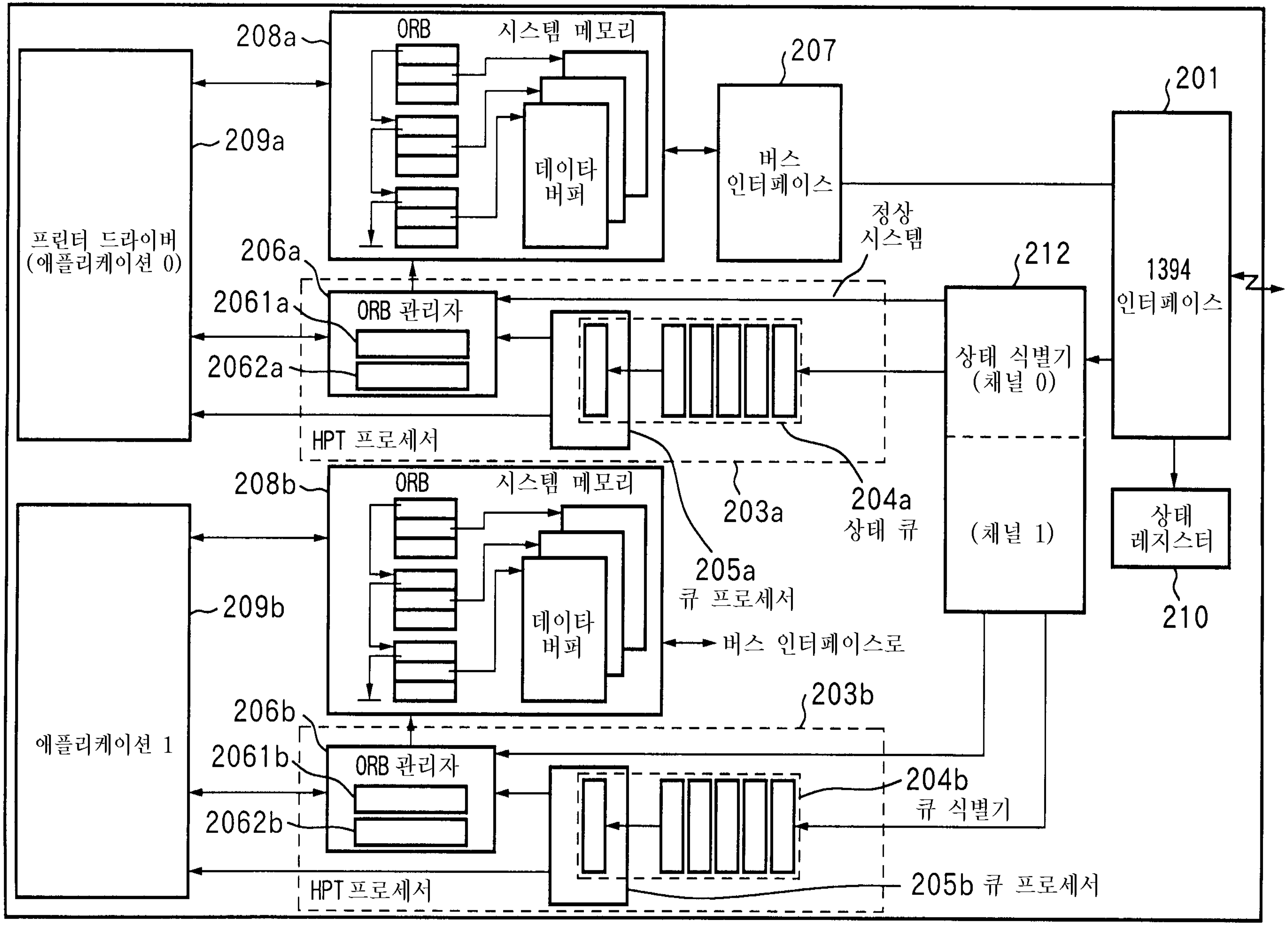
도면 21


HPT
SBP-2
1394
1394
SBP-2
HPT 페치 에이전트
실행 에이전트
Unsolicited Status Block 를 큐에 카피
Unsolicited Status Block 전송
2103
Unsolicited Status Block (직접)
2102
Unsolicited Status블랙 생성 빈 큐 1씩 감소
2101
데이타 생성
Unsolicited Status Block 전송 요청
빈 큐를 1씩 감소
2105
ORB 링크
2106
2107
큐의 선두 Unsolicited Status Block 데이타 판독
ORB 링크 요청
2104
도어벨 킥 요청
도어벨 기록
도어벨 킥 메시지
ORB 기록
2109
ORB 판독 요청
2108
타겟
개시전
ORB (판독 요청)
ORB
ORB 큐 속성 해석
2110
해당 QUE에 저장된 ORB 해석
2111
Status Block 생성
Status Block
Status Block 생성 전송 요청
빈 QUT를 1씩 증가하여 데이타 버퍼 삭제
링크 삭제
2112
빈 QUE를 1씩 증가
2113

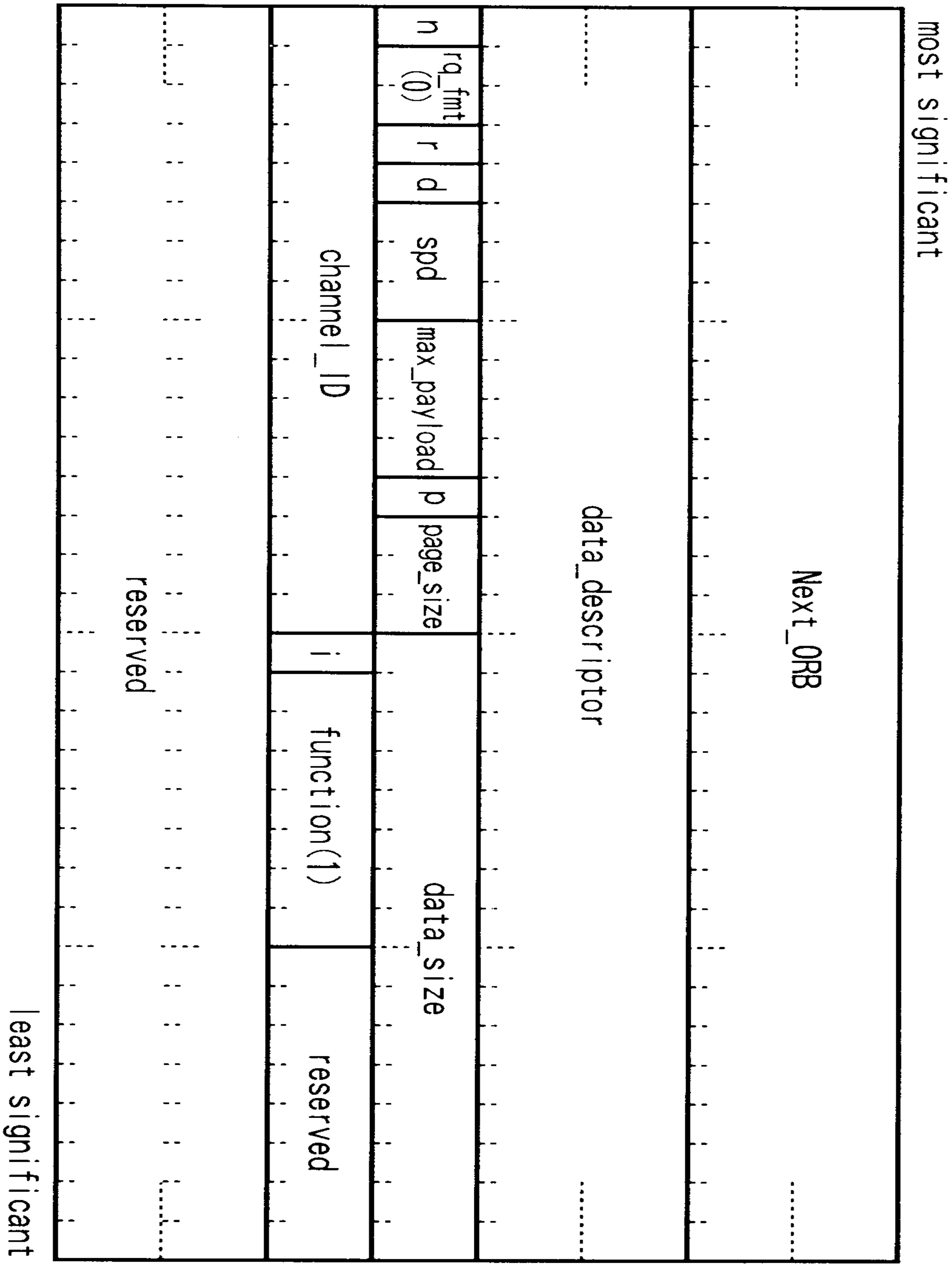
도면 22


101
1394 인터페이스
108
도어벨 레지스터
102
버스 인터페이스
명령 페치 에이전트 (채널 0)
(채널 1)
113
큐잉 명령 (ORB)
직접 명령 (ORB)
104a
프리페치 큐
큐잉 명령 (ORB)
직접 명령 (ORB)
104b
프리페치 큐
105a
큐잉 실행 에이전트
106a
즉시 실행 에이전트
105b
큐잉 실행 에이전트
106b
즉시 실행 에이전트
데이타 프로세서 0
107a
데이타 프로세서 1
107b

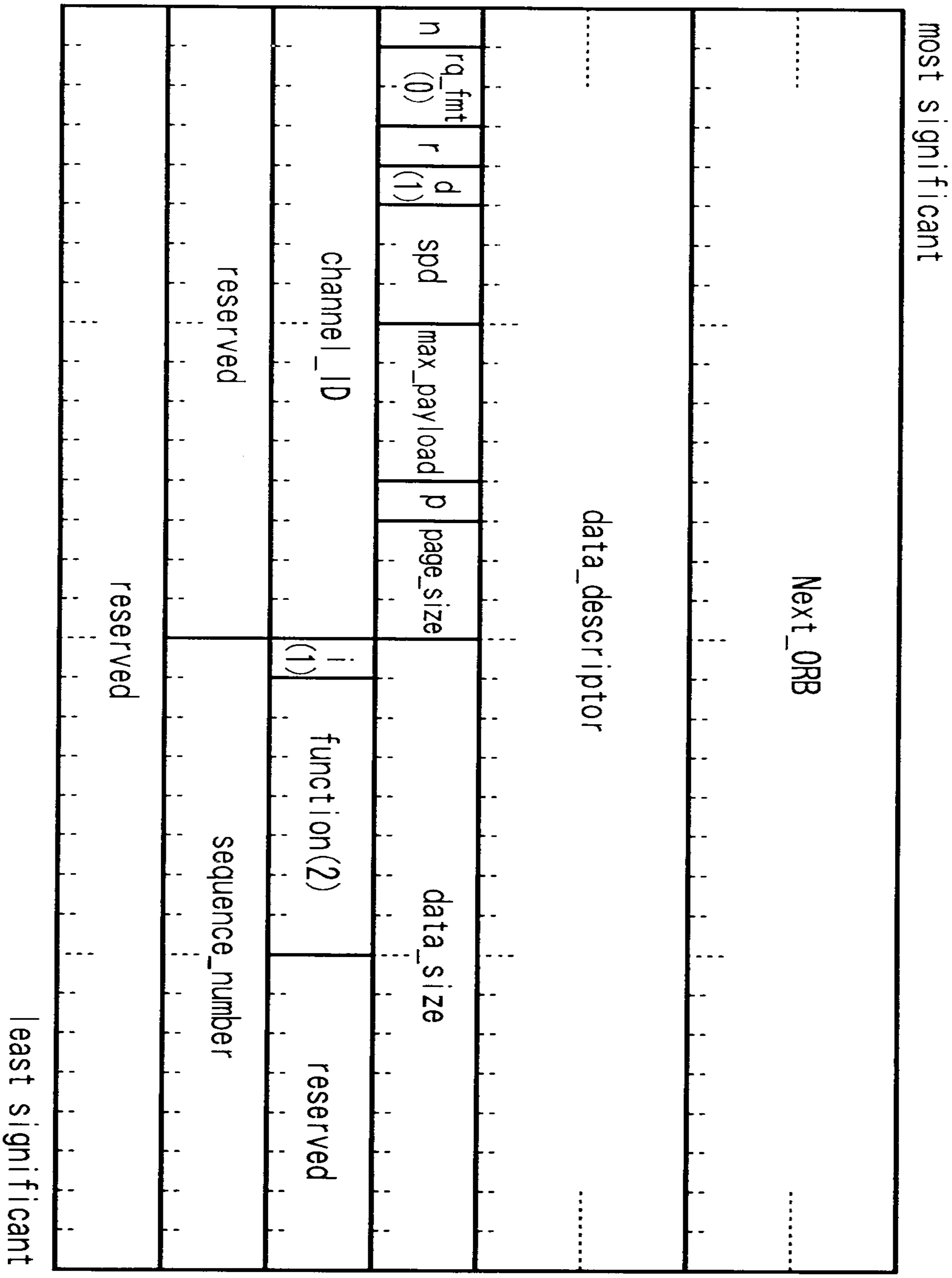
도면 23


208a
209a
프린터 드라이버
(애플리케이션 0)
ORB
시스템 메모리
데이터
버퍼
207
버스
인터페이스
201
1394
인터페이스
정상
시스템
212
206a
2061a
2062a
ORB 관리자
HPT 프로세서
상태 식별기
(채널 0)
(채널 1)
203a
204a
상태 큐
205a
큐 프로세서
상태
레지스터
210
208b
209b
ORB
시스템 메모리
데이터
버퍼
버스 인터페이스로
203b
애플리케이션 1
206b
2061b
2062b
ORB 관리자
204b
큐 식별기
HPT 프로세서
205b 큐 프로세서

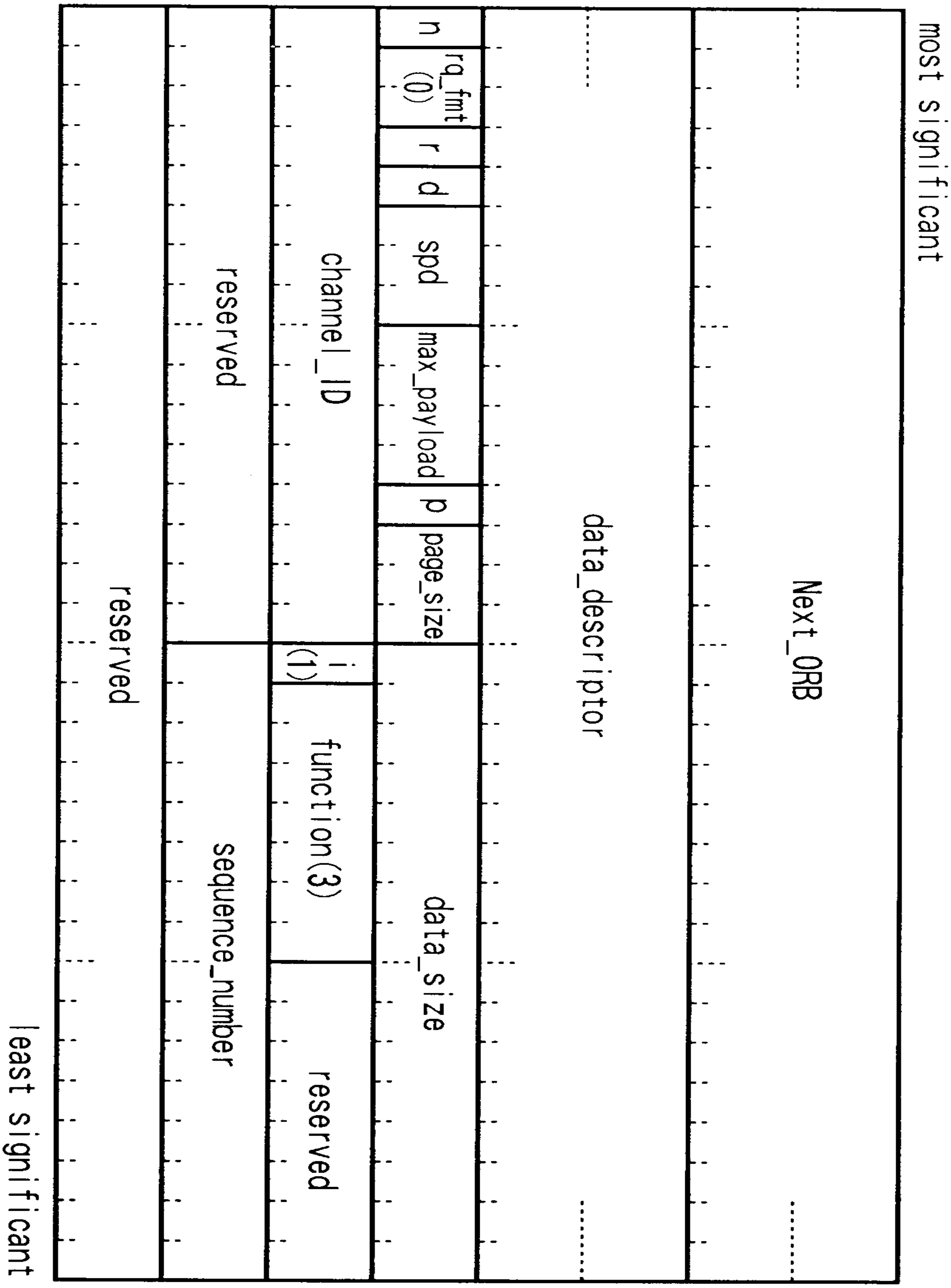
도면 24


most significant
Next_ORB
data_descriptor
n
rq_fmt (0)
r
d
spd
max_payload
p
page_size
data_size
channel_ID
i
function(1)
reserved
reserved
least significant

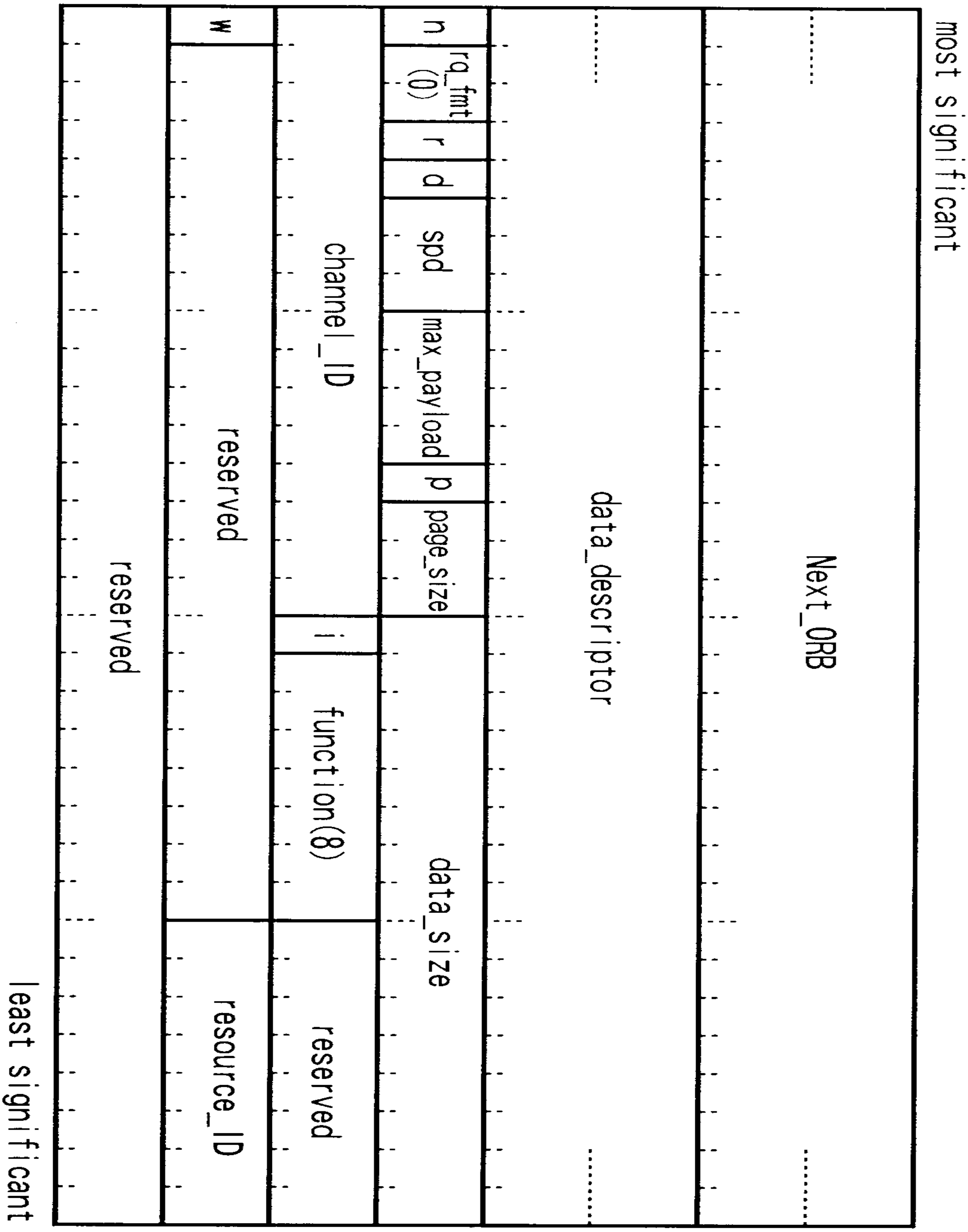
도면 25


most significant
Next_ORB
data_descriptor
n
rq_fmt (0)
r
d (1)
spd
max_payload
p
page_size
data_size
channel_ID
j (1)
function(2)
reserved
reserved
sequence_number
reserved
least significant

도면 26


most significant
Next_ORB
data_descriptor
n
rq_fmt (0)
r
d
spd
max_payload
p
page_size
data_size
channel_ID
i (1)
function(3)
reserved
sequence_number
reserved
reserved
least significant

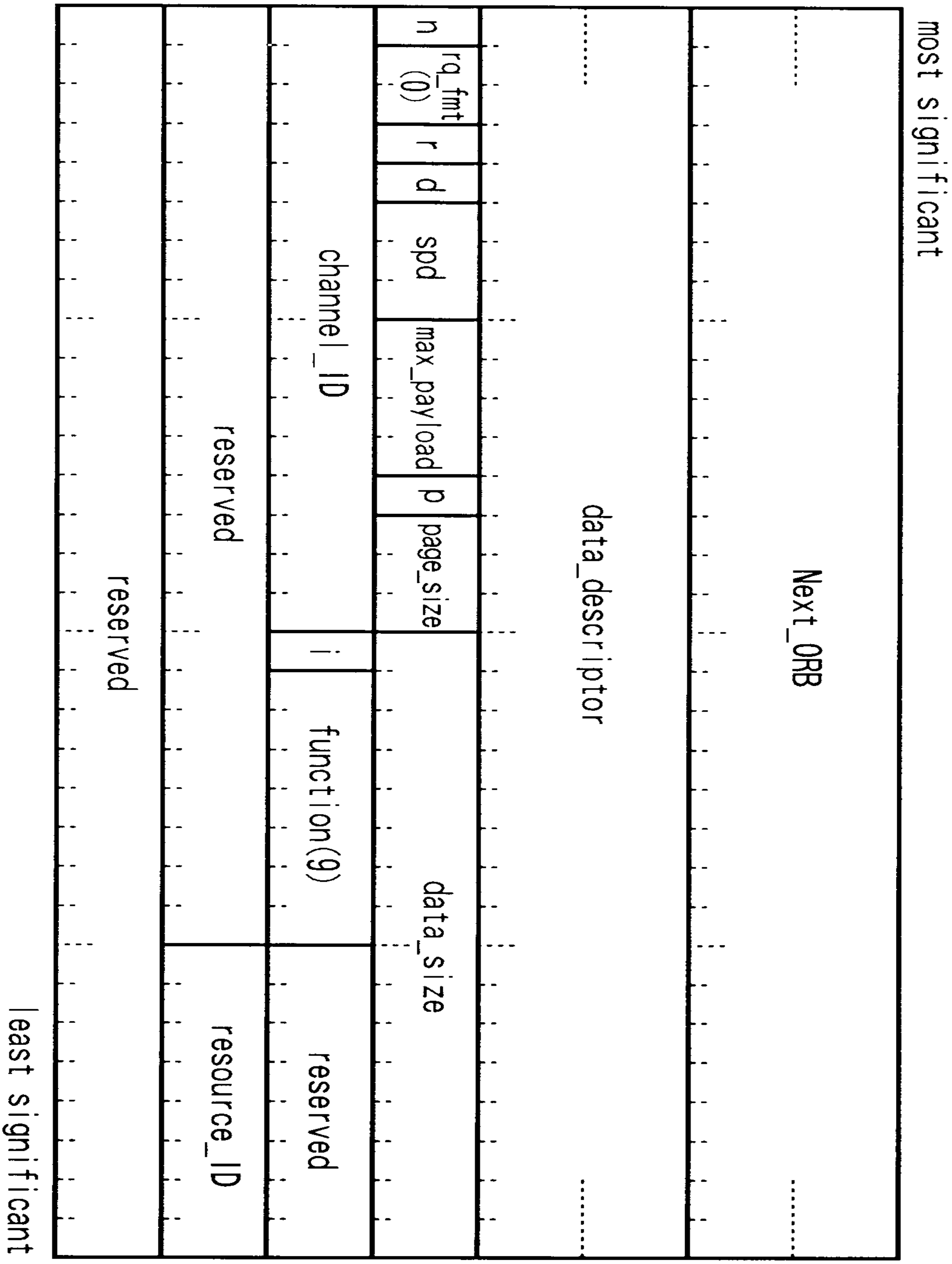
도면 27a


most significant
Next_ORB
data_descriptor
n
rq_fmt
(0)
r
d
spd
max_payload
p
page_size
data_size
channel_ID
i
function(8)
reserved
w
reserved
resource_ID
reserved
least significant

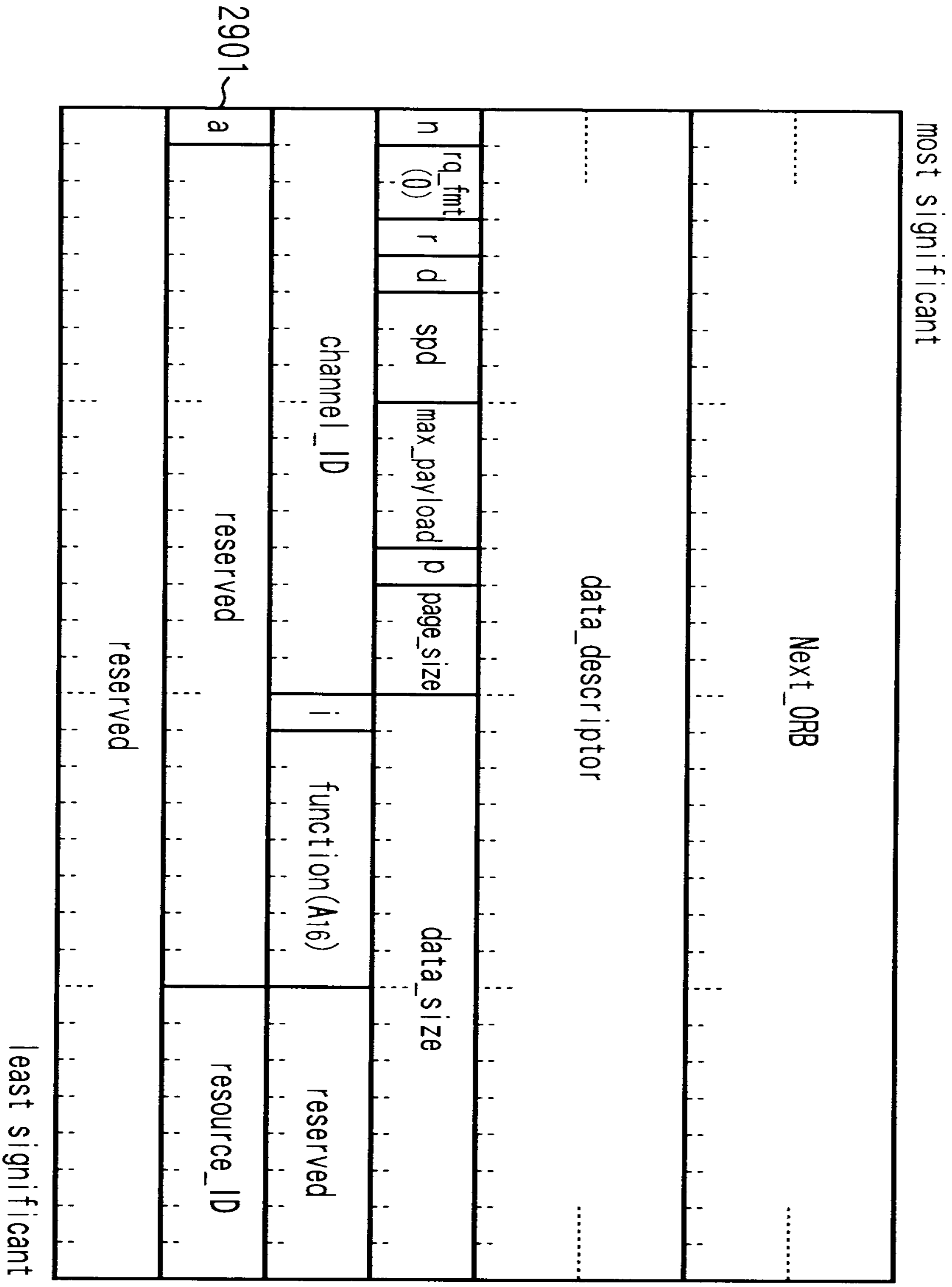
도면 27b

Value	Resource
0	device class and logical unit characteristic dependent
1-F_{16}	Reserved for future standardization
10_{16}-$7F_{16}$	Control set dependent

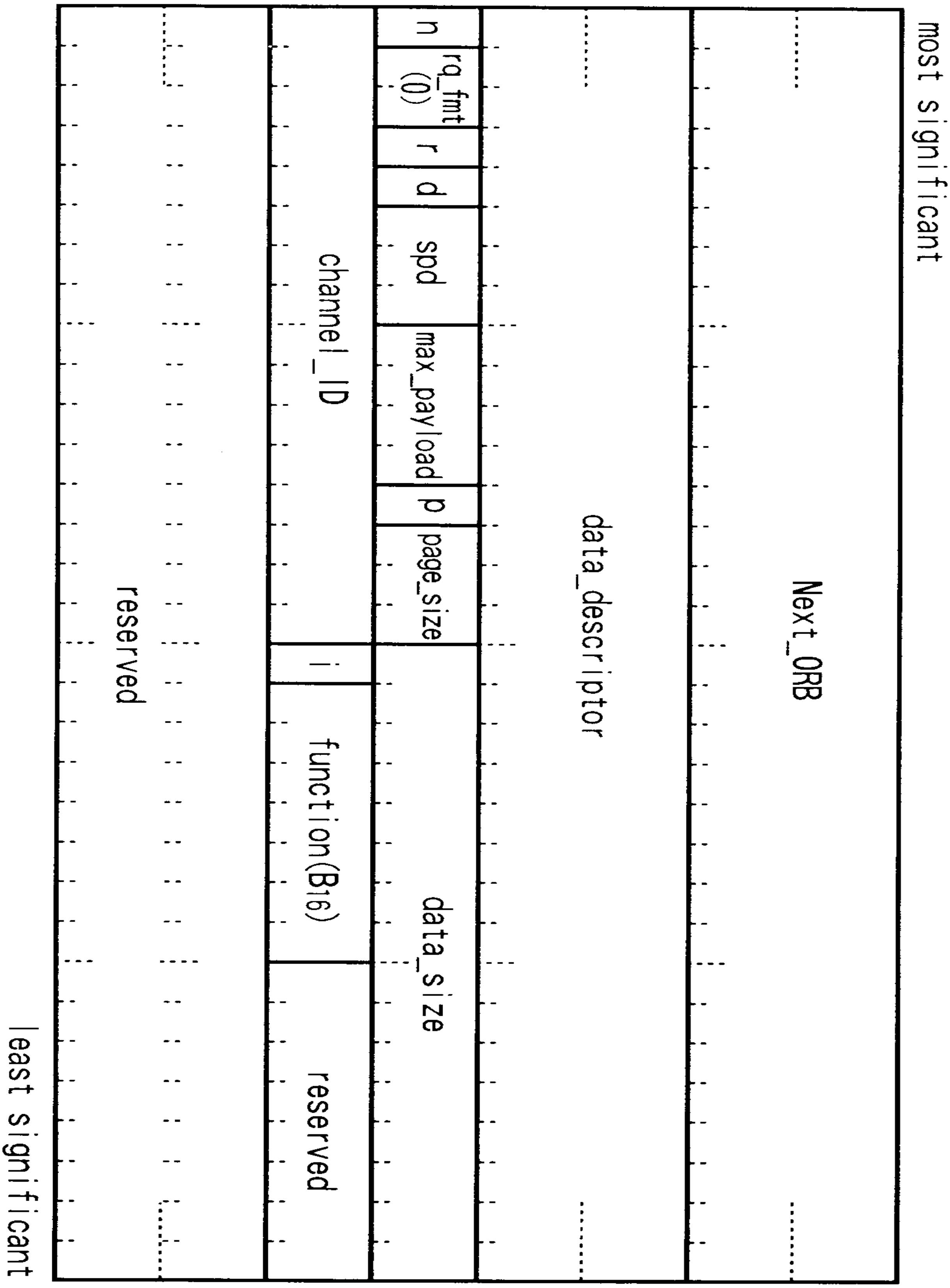
도면 28


most significant
Next_ORB
data_descriptor
n
rq_fmt
(0)
r
d
spd
max_payload
p
page_size
data_size
channel_ID
i
function(9)
reserved
reserved
resource_ID
reserved
least significant

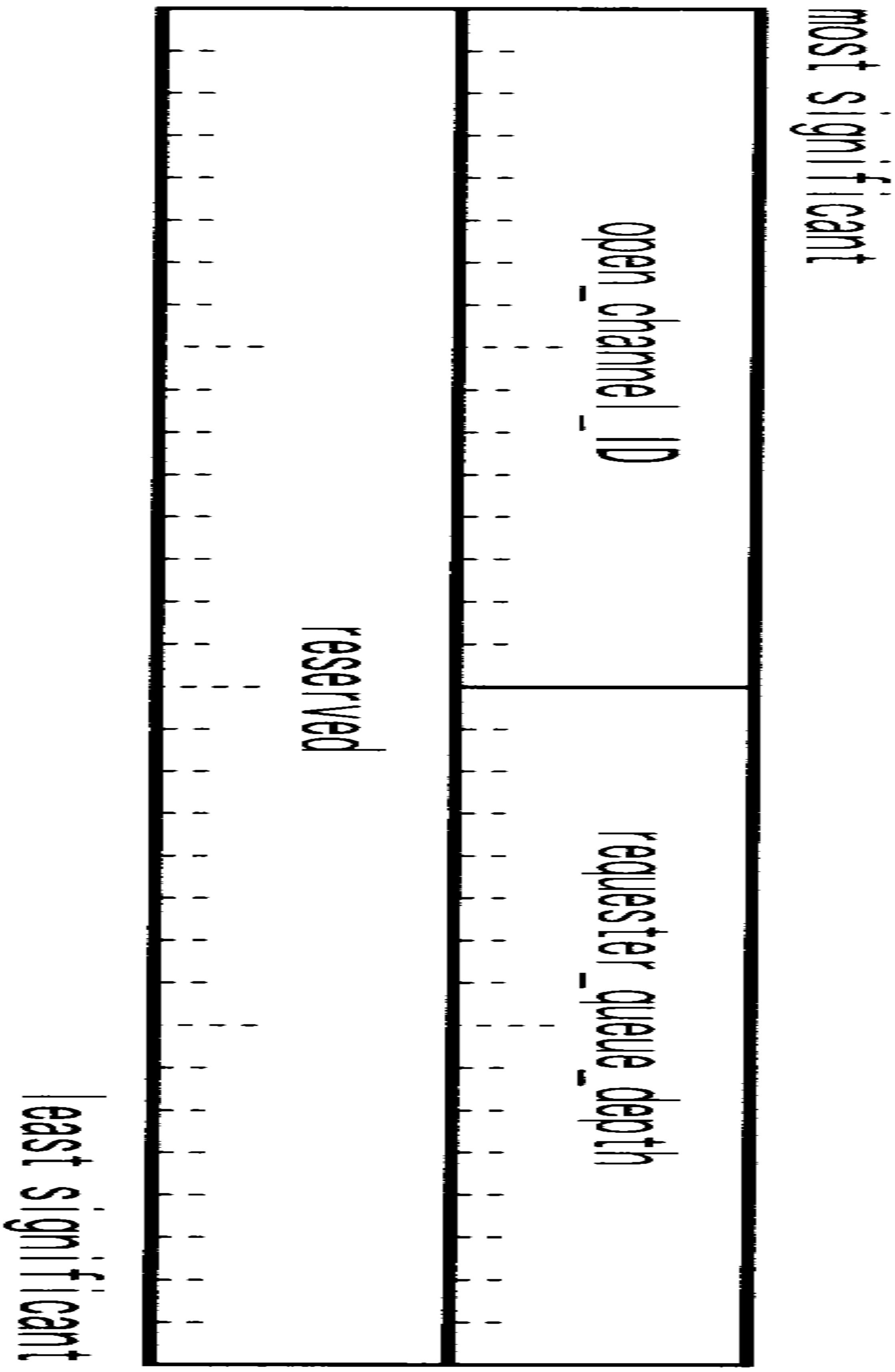
도면 29


most significant
Next_ORB
data_descriptor
n
rq_fmt (0)
r
d
spd
max_payload
p
page_size
data_size
channel_ID
i
function(A16)
reserved
2901
a
reserved
resource_ID
reserved
least significant

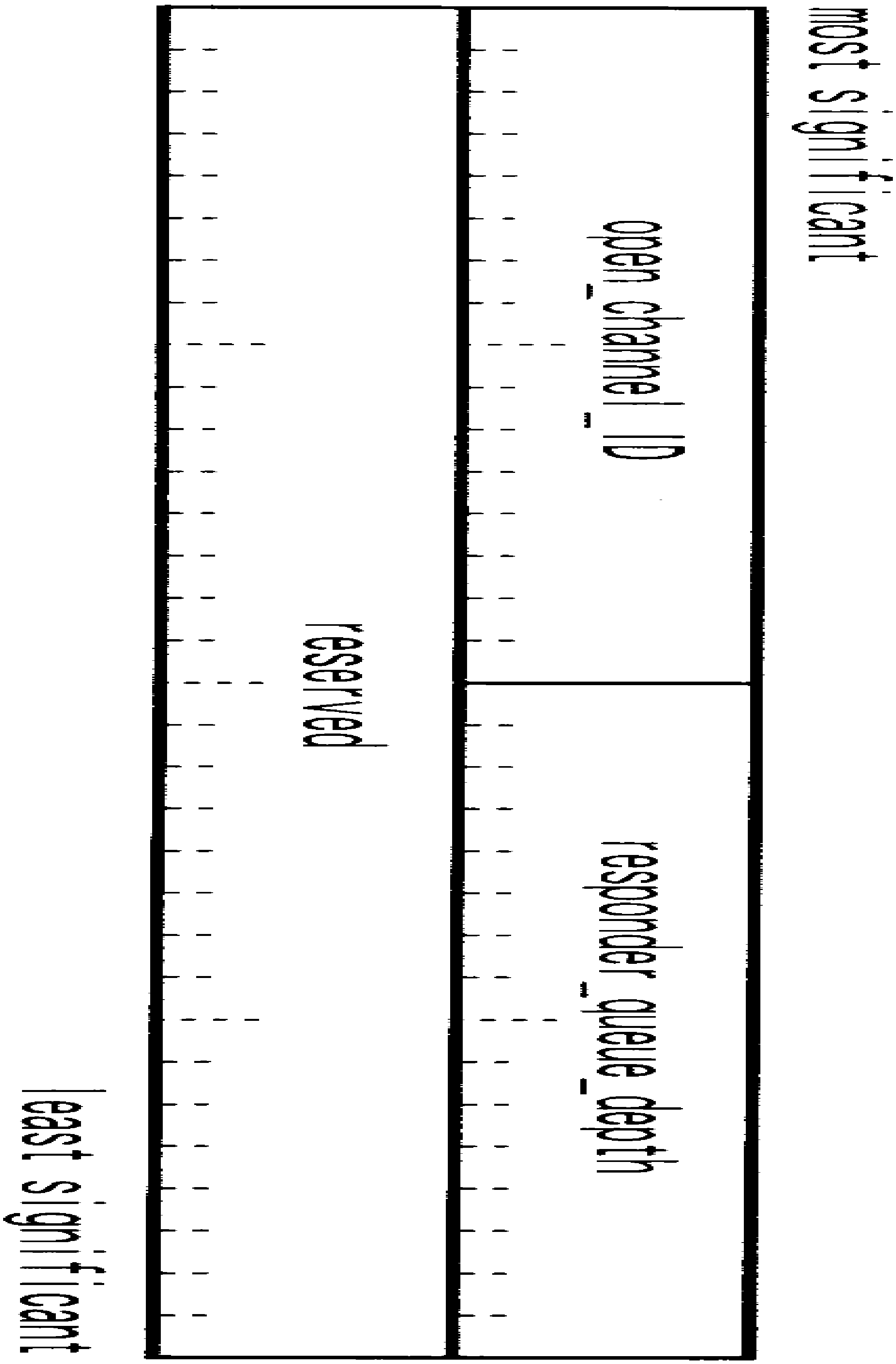
도면 30


most significant
Next_ORB
data_descriptor
n
rq_fmt (0)
r
d
spd
max_payload
p
page_size
data_size
channel_ID
i
function(B16)
reserved
reserved
least significant

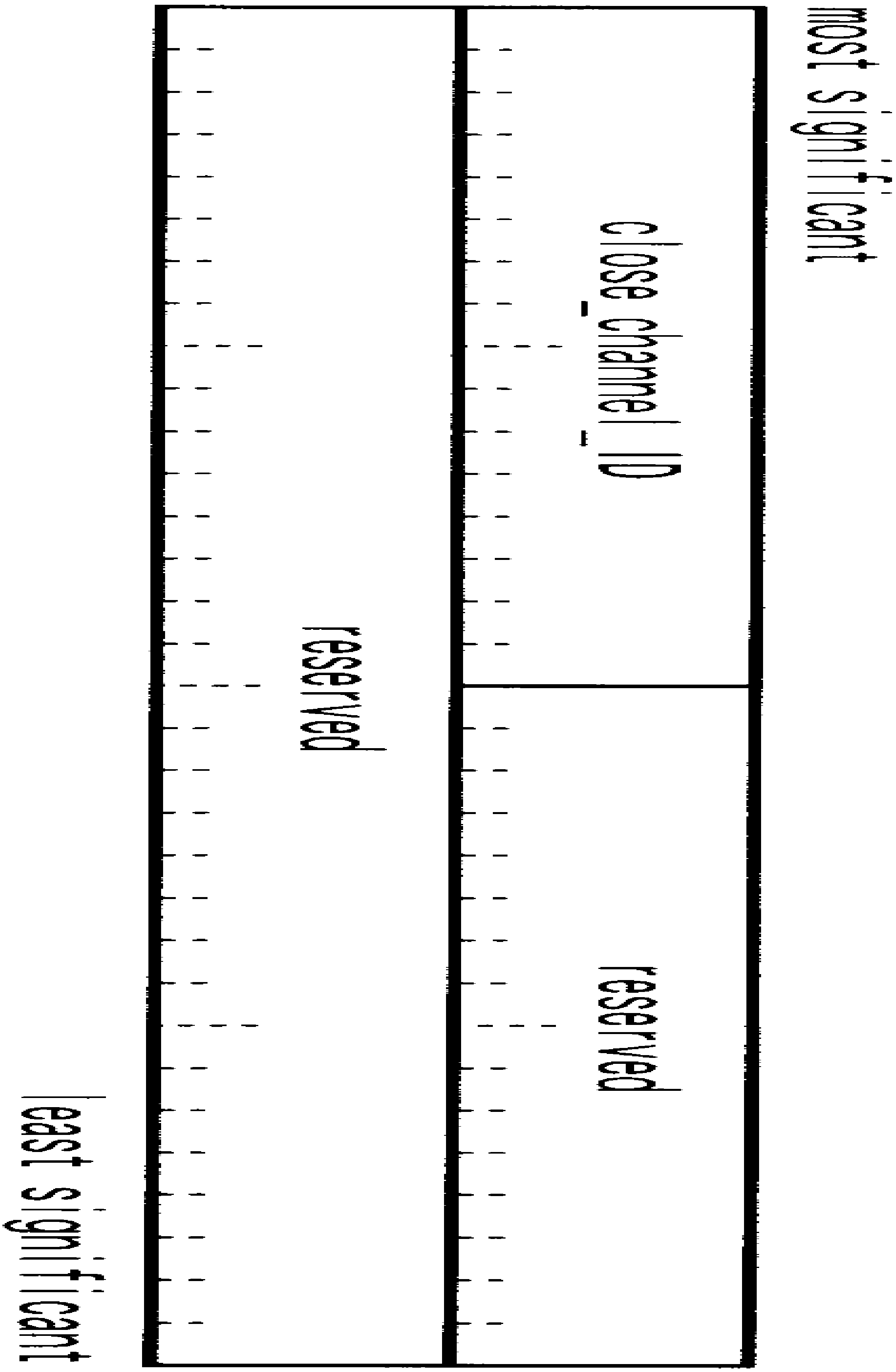
도면 31a


most significant
open_channel_ID
requester_queue_depth
reserved
least significant

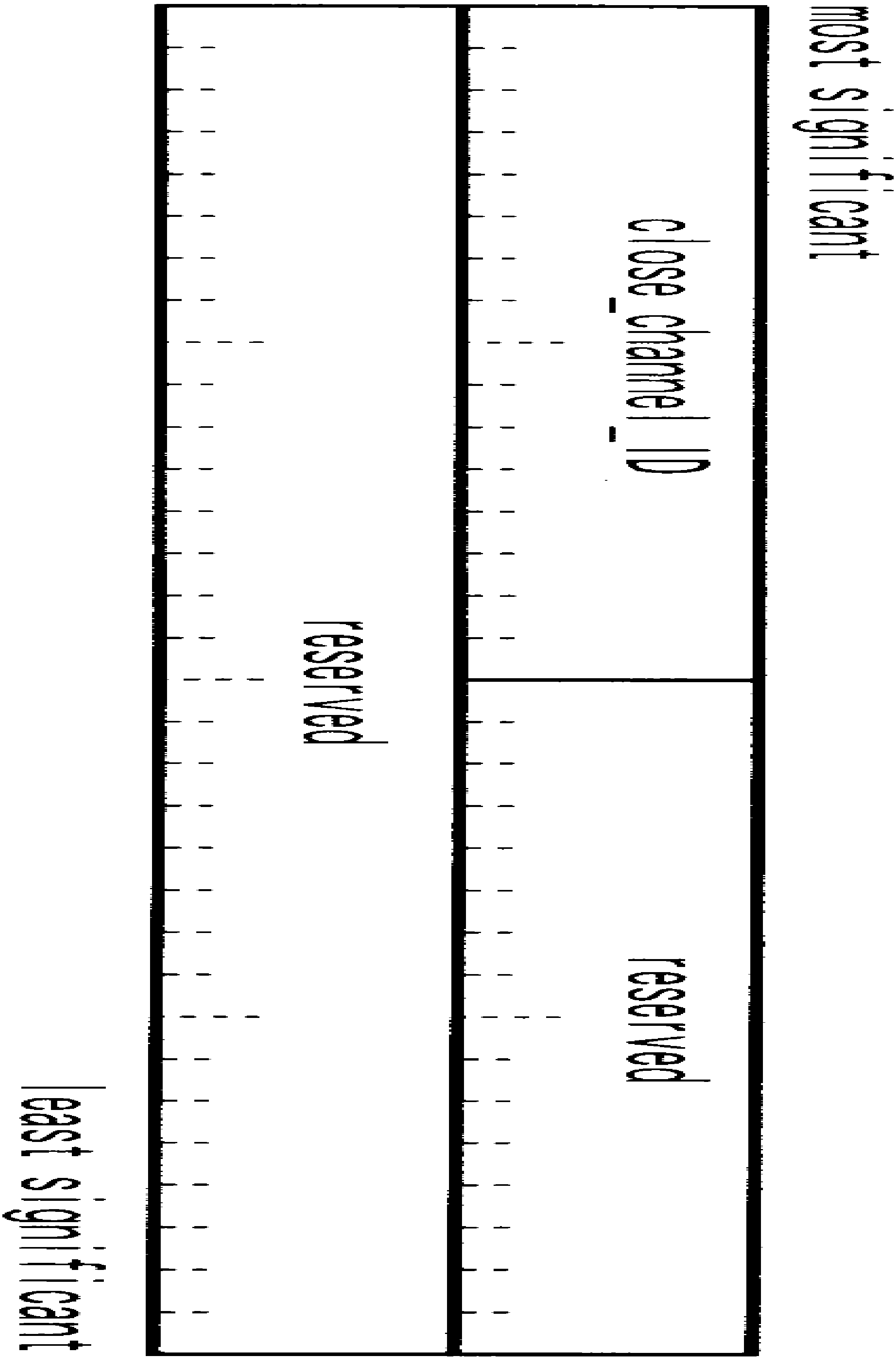
도면 31b


most significant
open_channel_ID
responder_queue_depth
reserved
least significant

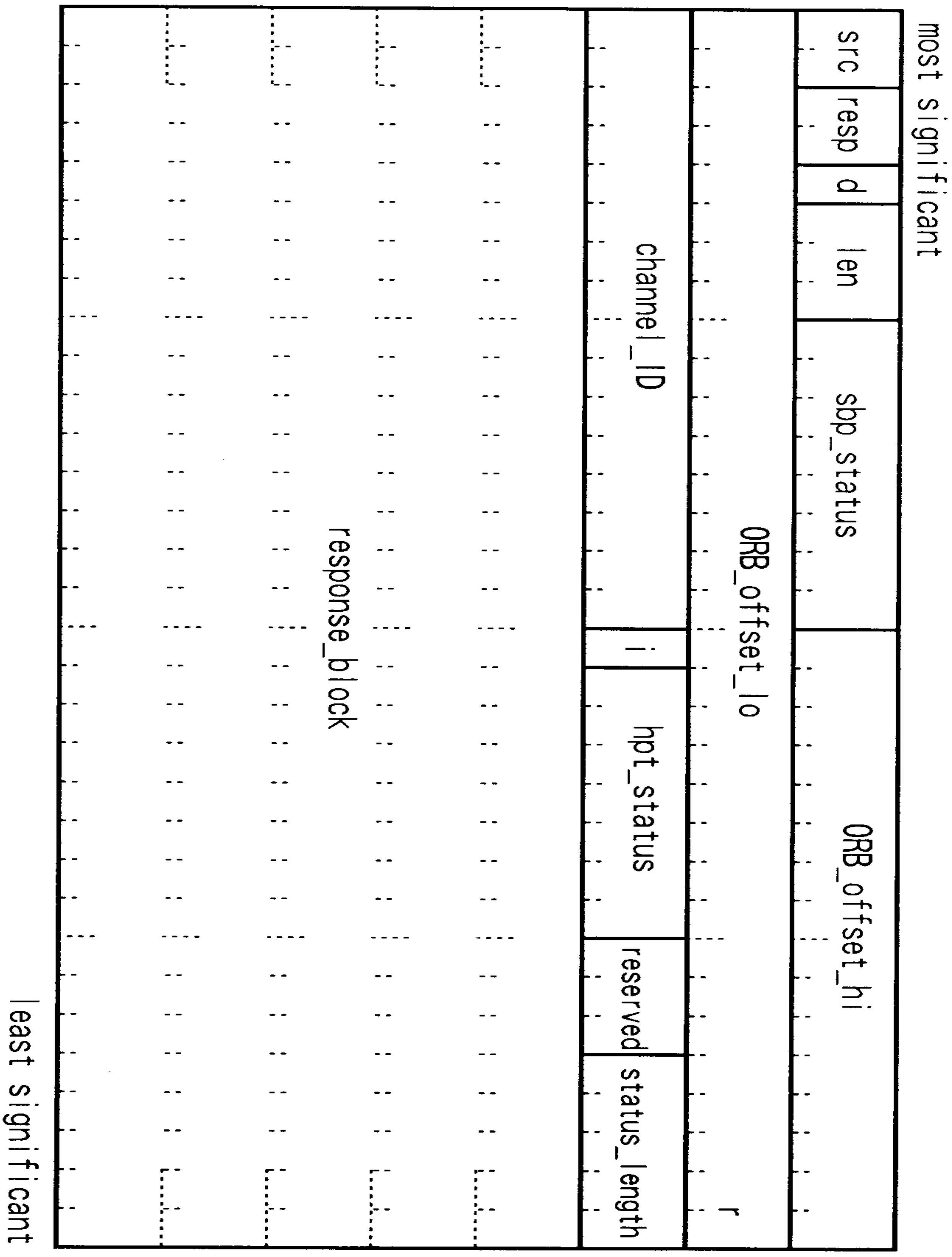
도면 32a


most significant
close_channel_ID
reserved
reserved
least significant

도면 32b


most significant
close_channel_ID
reserved
reserved
least significant

도면 33a


most significant
src
resp
d
len
sbp_status
ORB_offset_hi
ORB_offset_lo
r
channel_ID
i
hpt_status
reserved
status_length
response_block
least significant

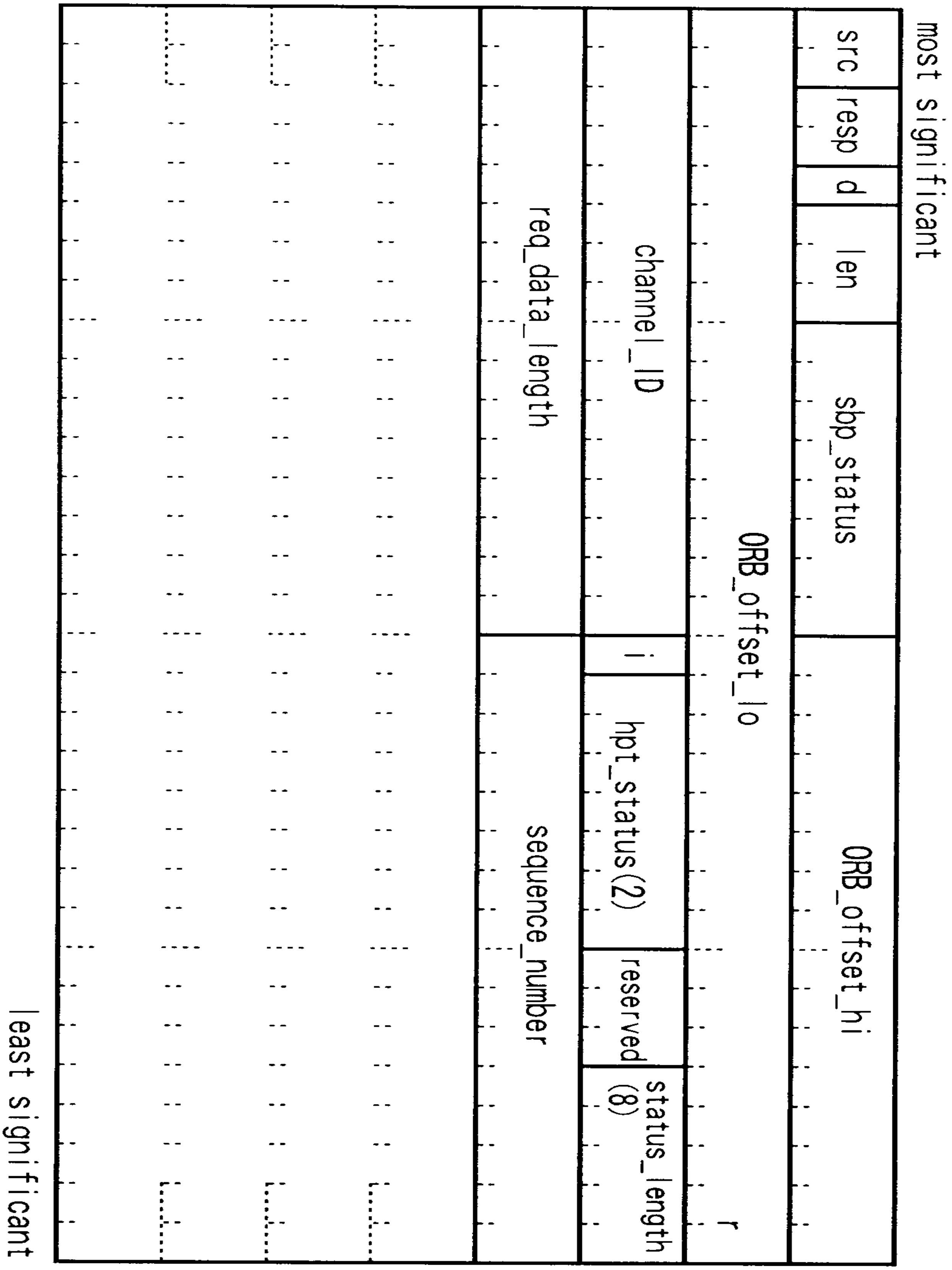
도면 33b

값	HPT 상태
0	앞으로의 표준화에 대비
1	앞으로의 표준화에 대비
2	판독 요청
3	직접 상태
4-7	앞으로의 표준화에 대비

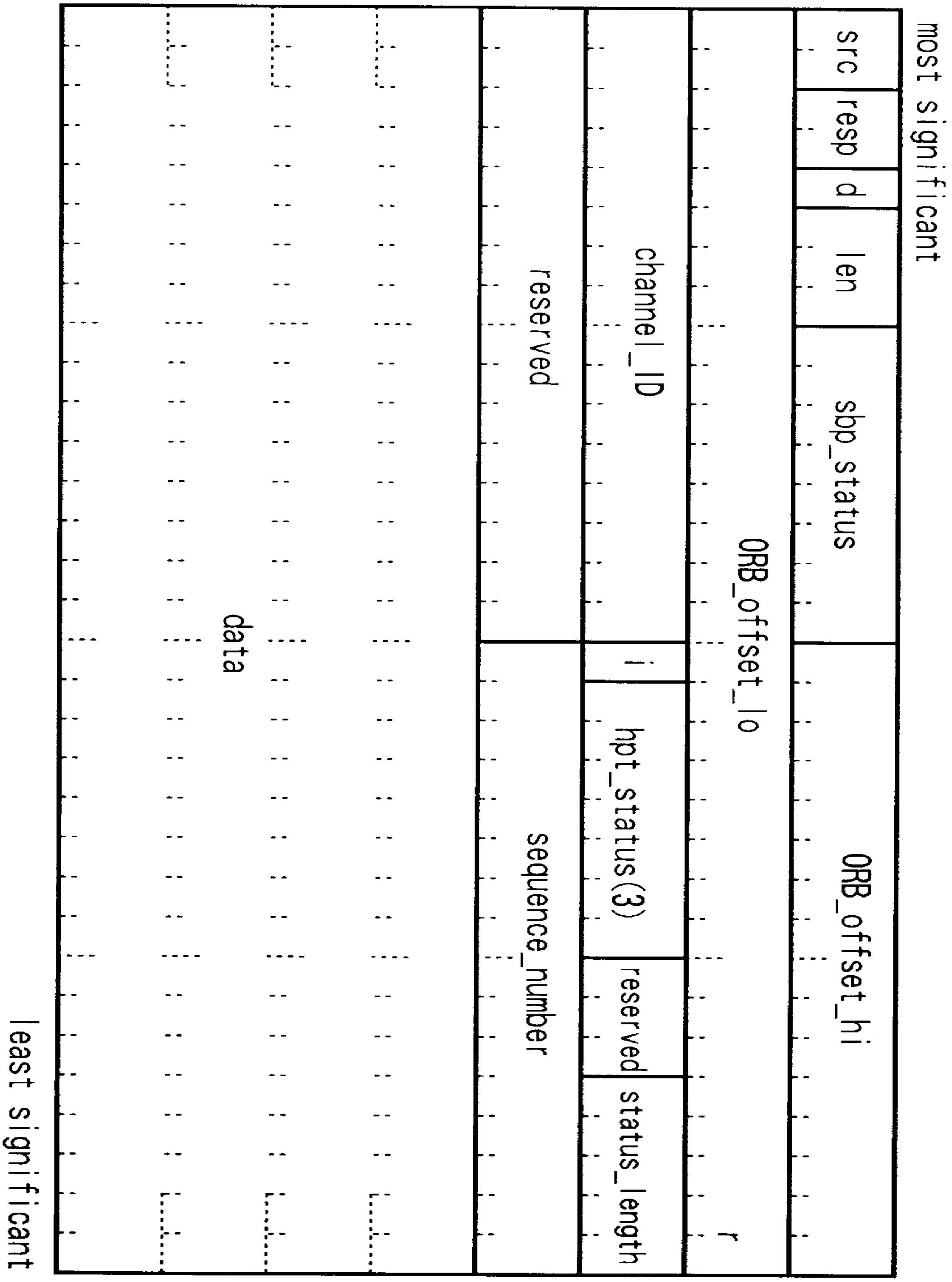
도면 33c

값	HPT 상태
8	디바이스 리소스 응답 획득
9	앞으로의 표준화에 대비
A_{16}	포기 디바이스 리소스 요청
B_{16}	기본 디바이스 상태
C_{16}-F_{16}	앞으로의 표준화에 대비
10_{16}-$6F_{16}$	제어 세트 의존

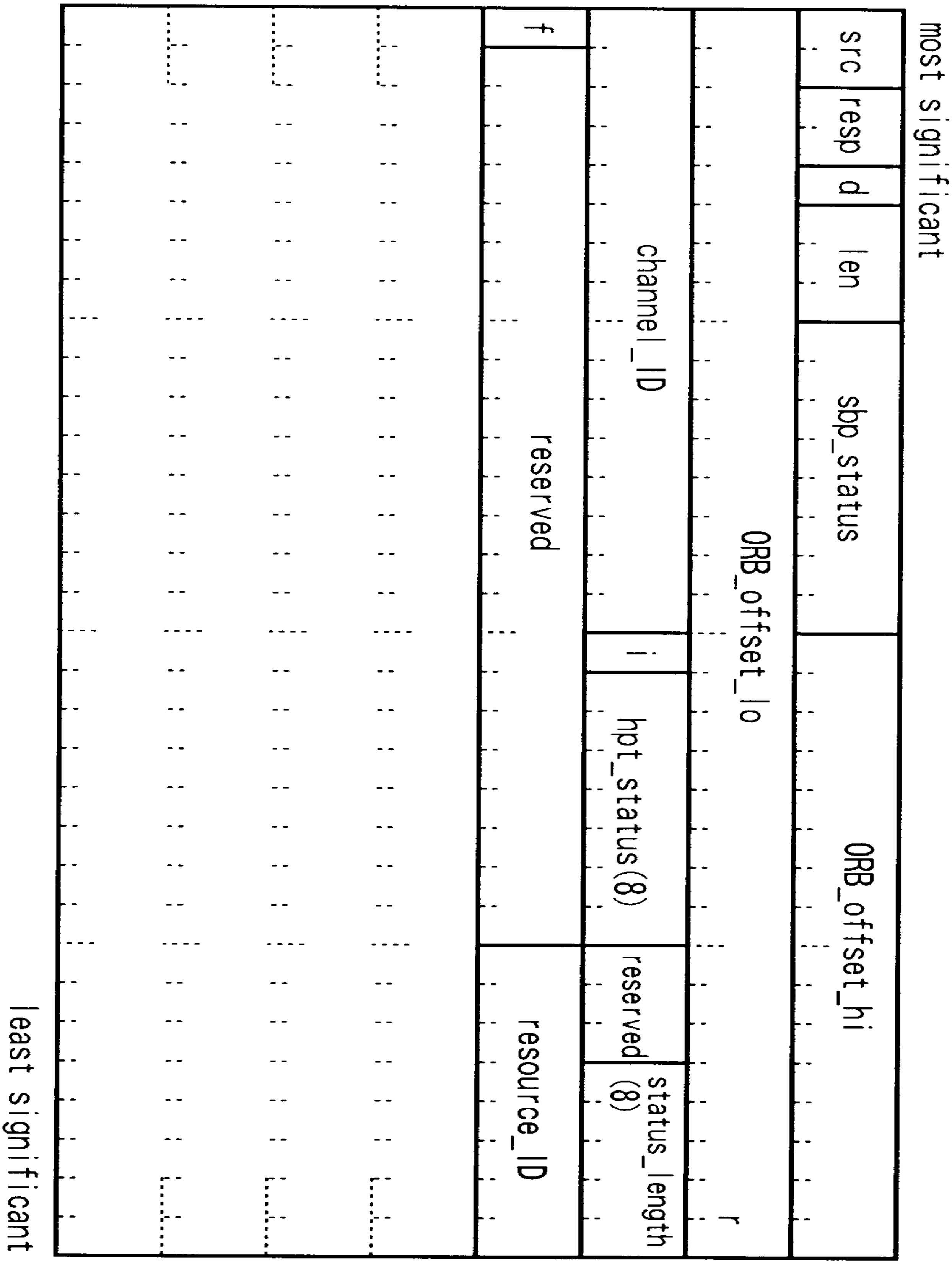
도면 34


most significant
src
resp
d
len
sbp_status
ORB_offset_hi
ORB_offset_lo
r
channel_ID
i
hpt_status(2)
reserved
status_length (8)
req_data_length
sequence_number
least significant

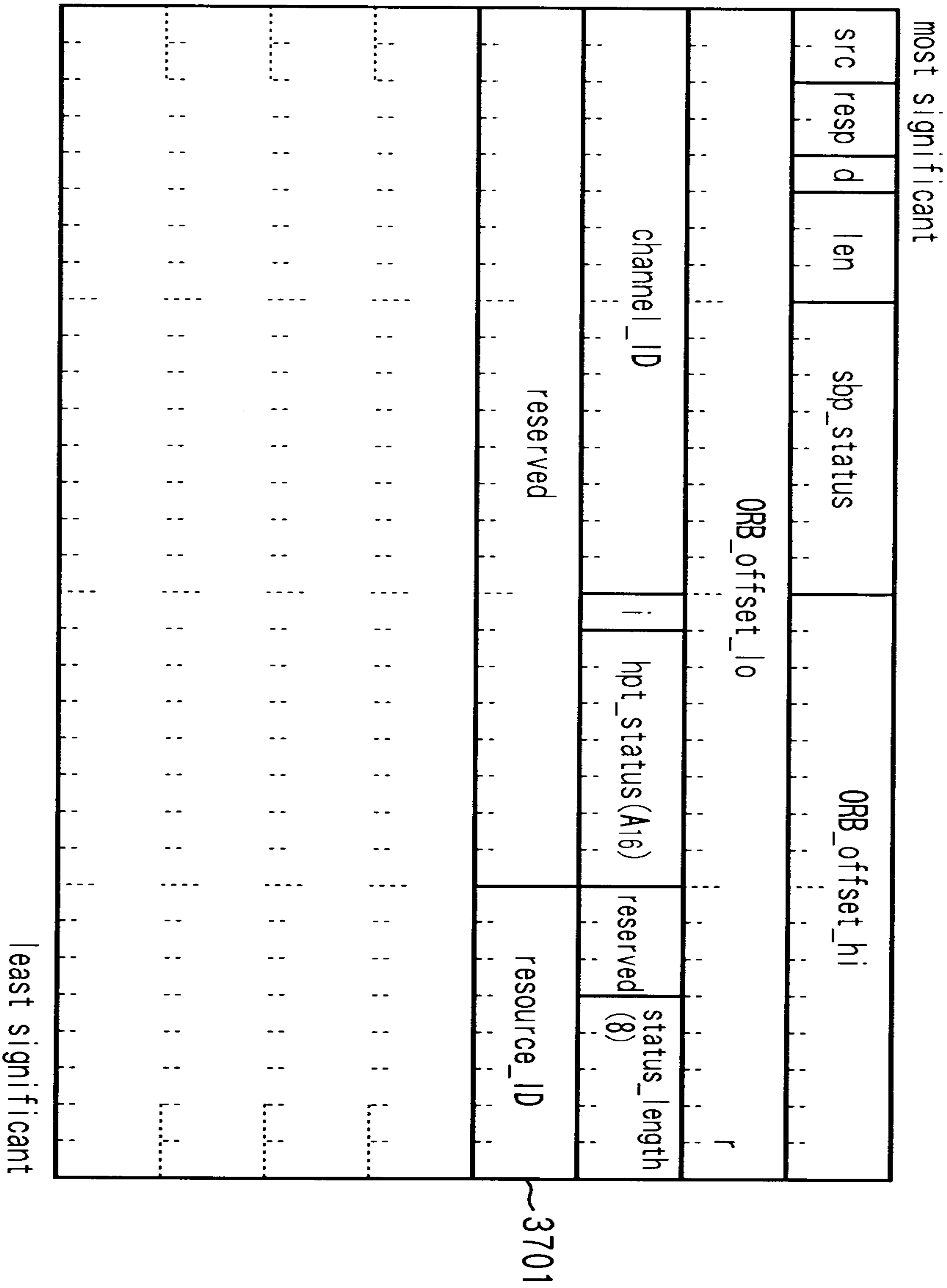
도면 35


most significant
src
resp
d
len
sbp_status
ORB_offset_hi
ORB_offset_lo
r
channel_ID
i
hpt_status(3)
reserved
status_length
reserved
sequence_number
data
least significant

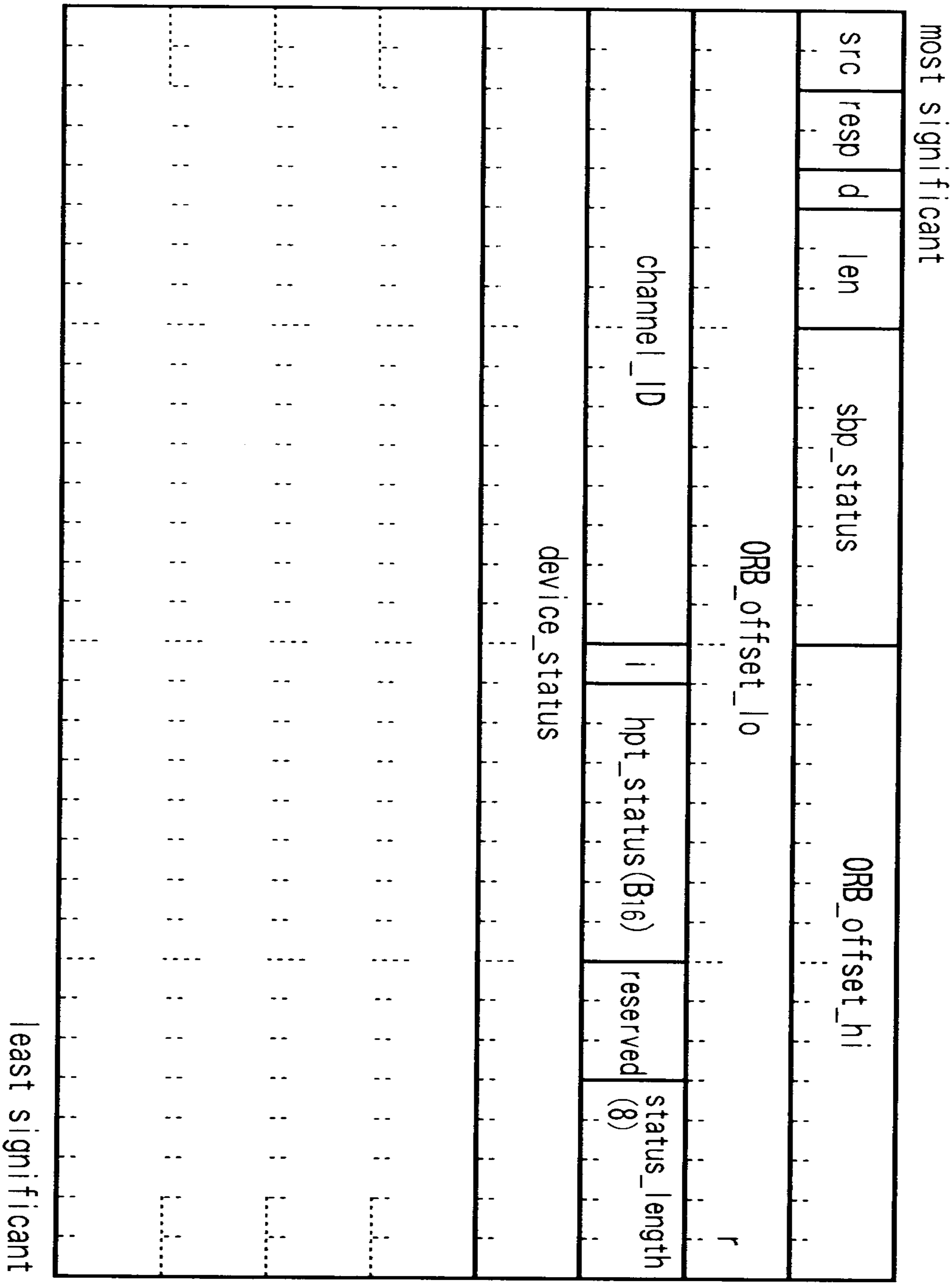
도면 36


most significant
src
resp
d
len
sbp_status
ORB_offset_hi
ORB_offset_lo
r
channel_ID
i
hpt_status(8)
reserved
status_length (8)
f
reserved
resource_ID
least significant

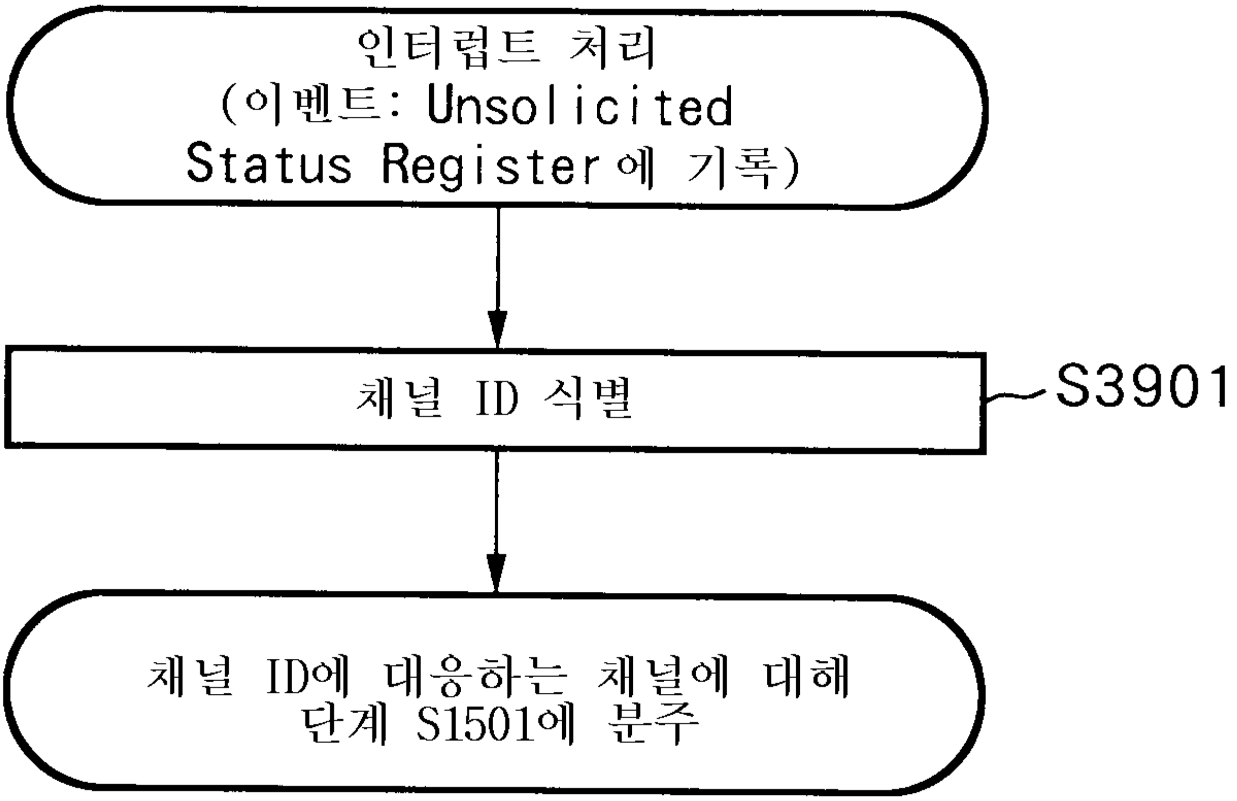
도면 37


most significant
src
resp
d
len
sbp_status
ORB_offset_hi
ORB_offset_lo
r
channel_ID
i
hpt_status(A16)
reserved(8)
status_length
reserved
resource_ID
least significant
3701

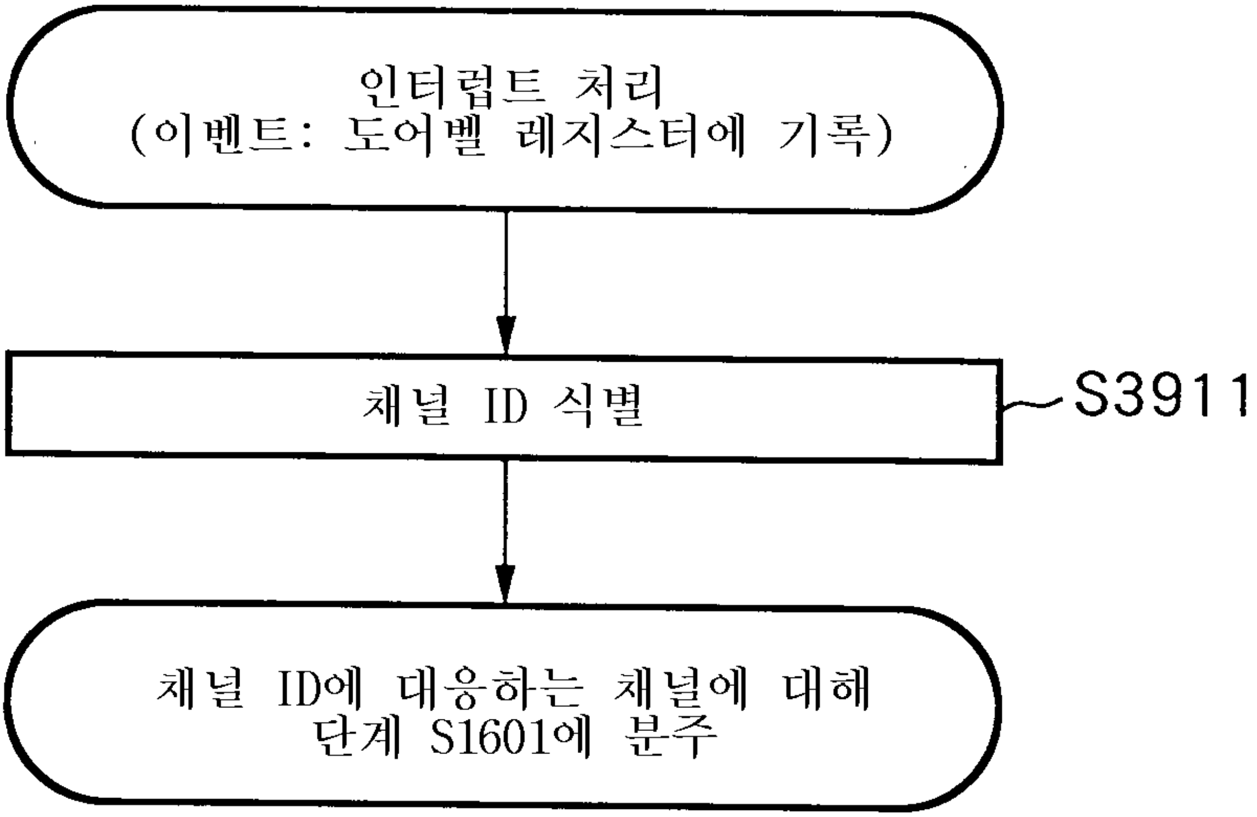
도면 38


most significant
src
resp
d
len
sbp_status
ORB_offset_hi
ORB_offset_lo
r
channel_ID
i
hpt_status(B16)
reserved
status_length
(8)
device_status
least significant

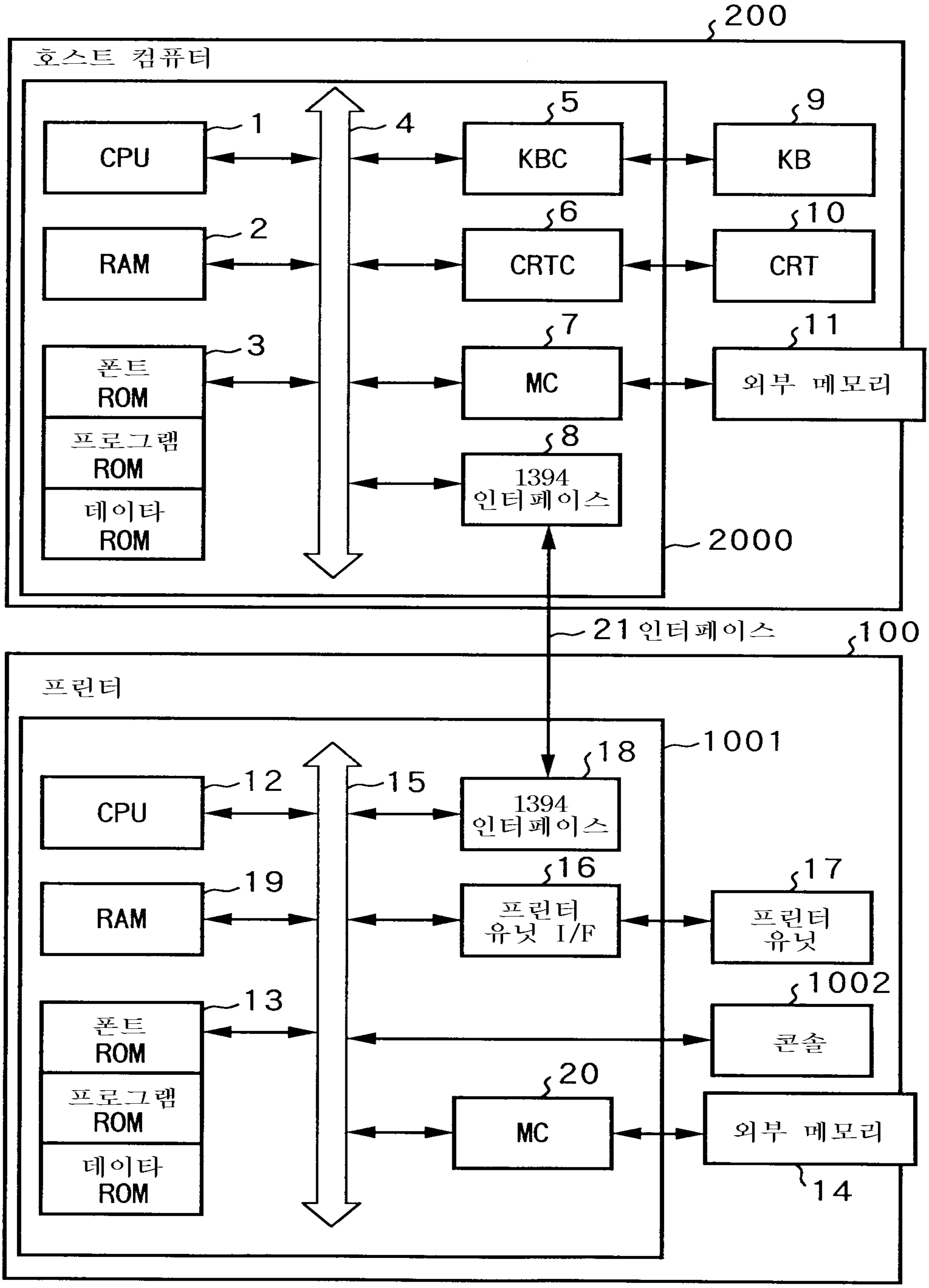
도면 39a


인터럽트 처리
(이벤트: Unsolicited
Status Register에 기록)
채널 ID 식별
S3901
채널 ID에 대응하는 채널에 대해
단계 S1501에 분주

도면 39b


인터럽트 처리
(이벤트: 도어벨 레지스터에 기록)
채널 ID 식별
S3911
채널 ID에 대응하는 채널에 대해
단계 S1601에 분주

도면 40


200
호스트 컴퓨터
CPU
1
RAM
2
폰트 ROM
프로그램 ROM
데이타 ROM
3
4
5
KBC
6
CRTC
7
MC
8
1394 인터페이스
9
KB
10
CRT
11
외부 메모리
2000
21 인터페이스
100
프린터
1001
CPU
12
15
18
1394 인터페이스
RAM
19
16
프린터 유닛 I/F
17
프린터 유닛
13
폰트 ROM
프로그램 ROM
데이타 ROM
1002
콘솔
20
MC
외부 메모리
14