



**(19) 대한민국특허청(KR)**  
**(12) 등록특허공보(B1)**

(45) 공고일자 2009년08월24일  
(11) 등록번호 10-0913658  
(24) 등록일자 2009년08월17일

(51) Int. Cl.  
G06F 9/44 (2006.01)  
(21) 출원번호 10-2004-7000718  
(22) 출원일자 2002년07월23일  
심사청구일자 2007년07월19일  
(85) 번역문제출일자 2004년01월16일  
(65) 공개번호 10-2004-0022460  
(43) 공개일자 2004년03월12일  
(86) 국제출원번호 PCT/IB2002/002889  
(87) 국제공개번호 WO 2003/012639  
국제공개일자 2003년02월13일  
(30) 우선권주장  
09/917,026 2001년07월26일 미국(US)  
(뒷면에 계속)  
(56) 선행기술조사문헌  
EP0889405 A  
WO2000/073912 A  
WO2000/074412 A  
US05938766 A

(73) 특허권자  
키오세라 와이어리스 코퍼레이션  
미국, 캘리포니아 92121, 샌 디에고, 캠퍼스 포인  
트 드라이브 10300  
(72) 발명자  
라자람, 고우리  
미국, 캘리포니아92122, 샌디에고, 아파트5330, 레본  
드라이브3520  
세컨도르프, 폴  
미국, 캘리포니아92129,  
샌디에고, 데이븐포트에비뉴14074  
카플란, 디에고  
미국, 캘리포니아92109,  
샌디에고, 솔다드마운튼로드5288  
(74) 대리인  
장명구, 강석용

전체 청구항 수 : 총 21 항

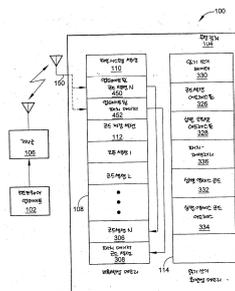
심사관 : 윤혜숙

**(54) 현장 업그레이드형 무선 통신 장치 소프트웨어 코드 섹션치밀화 시스템 및 방법**

**(57) 요약**

무선 통신 장치 메모리에 저장된 소프트웨어 명령을 재조직하기 위한 방법 및 시스템이 제공된다. 이 방법은, 1) 연속적으로 어드레싱되는 다수의 제 2 메모리 블록을 생성함으로써, 각각의 메모리 블록을 해당 코드 섹션으로 식별함으로써, 그리고 식별된 메모리 블록에 코드 섹션들을 저장함으로써, 해당 시작 어드레스에서 코드 섹션이 시작하는, 다수의 현 코드 섹션에 무선 장치 시스템 소프트웨어를 저장하고, 2) 무선 통신 장치 에어 인터페이스를 통해 새 코드 섹션을 수신하며, 3) 업데이트를 위해 현 코드 섹션을 식별하고, 4) 코드 섹션 크기를 연산하며, 5) 코드 섹션 크기 연산에 따라 치밀화 스케줄을 발생시키고, 6) 현 코드 섹션의 크기를 조정하며, 7) 현 코드 섹션의 크기 조정에 따라 코드 섹션 시작 어드레스를 변경시키고, 8) 코드 섹션들을 파일 시스템 섹션으로 일시적으로 이동시키며, 9) 치밀화 스케줄에 따라 연속 어드레싱을 관리하도록 파일 시스템 섹션으로부터의 상기 코드 섹션들을 메모리 블록에게로 저장함으로써 식별된 현 코드 섹션을 새 코드 섹션으로 대체하고, 그리고 10) 업데이트된 시스템 소프트웨어를 실행하는 단계들을 포함한다.

**대표도 - 도1**



(30) 우선권주장

09/916,460	2001년07월26일	미국(US)
09/916,900	2001년07월26일	미국(US)
09/927,131	2001년08월10일	미국(US)
09/969,305	2001년10월02일	미국(US)

---

## 특허청구의 범위

### 청구항 1

무선 통신 장치의 비휘발성 메모리(108)의 이산 블록(400, 402, 404, 406, 460)에 저장된 다수의 현 코드 섹션(current code section: 302, 304, 306, 308, 344) 중 하나 이상의 현 코드 섹션으로, 새 코드 섹션(new code section: 450, 452)을 치밀화(compacting)하기 위한 방법에 있어서, 상기 방법은

- 무선 통신 네트워크를 통해 소프트웨어 업데이트(102)를 수신하는 단계,
- 새 코드 섹션(450, 452)을 획득하기 위해 상기 소프트웨어 업데이트(102)를 파싱(parsing)하는 단계,
- 상기 새 코드 섹션(450, 452)의 크기를 판단하는 단계,
- 상기 새 코드 섹션을 저장하기 위해 상기 비휘발성 메모리(108)의 블록을 식별하는 단계로서, 이때 상기 비휘발성 메모리의 블록은 시작점 어드레스(start address)를 갖는 것이 특징인 식별하는 단계,
- 상기 새 코드 섹션을 상기 비휘발성 메모리의 식별된 블록으로 기록하는 단계로서, 이때 상기 새 코드 섹션은 상기 시작점 어드레스에서 시작하는 것이 특징인 기록하는 단계, 그리고
- 상기 새 코드 섹션을 위한 식별자와 시작점 어드레스를 포함하는 엔트리(entry)를 추가하기 위해, 코드 섹션 어드레스 테이블을 업데이트하는 단계로서, 이때 상기 식별자와 시작점 어드레스가 관련되어 있는 것이 특징인 업데이트하는 단계를 포함하되,

상기 식별하는 단계는,

- 상기 비휘발성 메모리의 새 블록을 할당하는 단계를 더 포함하고,

상기 할당하는 단계는,

- 상기 비휘발성 메모리의 다수의 현 코드 섹션(302, 304, 306, 308, 344) 중, 심벌 라이브러리 및 미사용 부분(340, 346)을 포함하는 현 코드 섹션을 식별하는 단계로서, 상기 현 코드 섹션은 상기 비휘발성 메모리(108)의 메모리 블록(400, 402, 404, 406, 460)의 시작점 어드레스(315, 320, 322, 324, 408, 410, 412, 414)에서 시작하고, 상기 미사용 부분(340, 346)은 상기 메모리 블록(400, 402, 404, 406, 460)의 종착지 어드레스에서 끝나는 것이 특징인 현 코드 섹션을 식별하는 단계,
- 상기 미사용 부분에 대한 시작점 어드레스를 결정하는 단계,
- 상기 미사용 부분에 대한 시작점 어드레스와 상기 메모리 블록의 종착지 어드레스 사이의 상기 비휘발성 메모리를 상기 새 코드 섹션(450, 452)을 저장하기 위한 새 메모리 블록으로 할당하는 단계,

를 포함하는 것을 특징으로 하는, 무선 통신 장치의 비휘발성 메모리의 이산 블록에 저장된 다수의 현 코드 섹션으로 새 코드 섹션을 치밀화하기 위한 방법.

### 청구항 2

제 1 항에 있어서, 상기 새 코드 섹션(450, 452)은 부분 코드 섹션(partial code section)을 포함하는 것을 특징으로 하는, 무선 통신 장치의 비휘발성 메모리의 이산 블록에 저장된 다수의 현 코드 섹션으로 새 코드 섹션을 치밀화하기 위한 방법.

### 청구항 3

제 2 항에 있어서, 상기 부분 코드 섹션은 심벌 라이브러리를 포함하는 것을 특징으로 하는, 무선 통신 장치의 비휘발성 메모리의 이산 블록에 저장된 다수의 현 코드 섹션으로 새 코드 섹션을 치밀화하기 위한 방법.

### 청구항 4

제 1 항에 있어서, 상기 판단하는 단계는, 상기 새 코드 섹션(450, 452)의 크기를 계산하는 단계를 더 포함하는 것을 특징으로 하는, 무선 통신 장치의 비휘발성 메모리의 이산 블록에 저장된 다수의 현 코드 섹션으로 새 코드 섹션을 치밀화하기 위한 방법.

**청구항 5**

제 1 항에 있어서, 상기 판단하는 단계는 상기 새 코드 섹션(450, 452)의 압축을 해제하는 단계 (decompressing)을 더 포함하는 것을 특징으로 하는, 무선 통신 장치의 비휘발성 메모리의 이산 블록에 저장된 다수의 현 코드 섹션으로 새 코드 섹션을 치밀화하기 위한 방법.

**청구항 6**

삭제

**청구항 7**

삭제

**청구항 8**

제 1 항에 있어서, 상기 시작점 어드레스를 결정하는 단계는 심벌 라이브러리로부터 심벌을 판독하는 단계를 포함하는 것을 특징으로 하는, 무선 통신 장치의 비휘발성 메모리의 이산 블록에 저장된 다수의 현 코드 섹션으로 새 코드 섹션을 치밀화하기 위한 방법.

**청구항 9**

제 8 항에 있어서, 상기 심벌은 현 코드 섹션에 대한 종착지 어드레스를 포함하는 것을 특징으로 하는, 무선 통신 장치의 비휘발성 메모리의 이산 블록에 저장된 다수의 현 코드 섹션으로 새 코드 섹션을 치밀화하기 위한 방법.

**청구항 10**

제 8 항에 있어서, 상기 심벌은 현 코드 섹션의 크기를 포함하는 것을 특징으로 하는, 무선 통신 장치의 비휘발성 메모리의 이산 블록에 저장된 다수의 현 코드 섹션으로 새 코드 섹션을 치밀화하기 위한 방법.

**청구항 11**

제 1 항에 있어서, 상기 기록하는 단계는 상기 비휘발성 메모리의 식별된 블록의 현 코드 섹션을 상기 새 코드 섹션(450, 452)으로 대체하는 단계를 더 포함하는 것을 특징으로 하는, 무선 통신 장치의 비휘발성 메모리의 이산 블록에 저장된 다수의 현 코드 섹션으로 새 코드 섹션을 치밀화하기 위한 방법.

**청구항 12**

삭제

**청구항 13**

삭제

**청구항 14**

삭제

**청구항 15**

삭제

**청구항 16**

삭제

**청구항 17**

삭제

**청구항 18**

삭제

**청구항 19**

삭제

**청구항 20**

삭제

**청구항 21**

삭제

**청구항 22**

무선 통신 장치(104)상에서, 새 코드 섹션(new code section; 450, 452)을 다수의 현 코드 섹션(current code section; 302, 304, 306, 308, 344)으로 치밀화(compacting)하기 위한 시스템에 있어서, 상기 시스템은

- 다수의 연속적으로 어드레스된 메모리 블록(400, 402, 404, 406)을 포함하는 비휘발성 메모리(108)로서, 각각의 메모리 블록은 시작점 어드레스(start address; 318, 320, 322, 324, 408, 410, 412, 414)와 종착점 어드레스(end address)를 포함하는 것이 특징인 상기 비휘발성 메모리,
- 상기 메모리 블록(400, 402, 404, 406)에 배열된 다수의 현 코드 섹션(302, 304, 306, 308, 344)으로서, 이때 각각의 코드 섹션은 상기 시작점 어드레스에서 시작하고, 해당 메모리 블록의 종착점 어드레스에서 종료하는 것이 특징인 상기 다수의 현 코드 섹션,
- 상기 현 코드 섹션(302, 304, 306, 308, 344)에 배열된 다수의 심벌 라이브러리(310, 312, 314, 316, 326, 328, 330, 332, 334, 336, 342)로서, 이때 상기 현 코드 섹션에는 각각 하나 이상의 심벌 라이브러리가 저장되는 것이 특징인 상기 다수의 심벌 라이브러리,
- 무선 통신 네트워크를 통해 소프트웨어 업데이트(102)를 수신하도록 구성된 에어 링크 인터페이스(air link interface; 150),
- 상기 비휘발성 메모리(108)의 제 1 블록(340, 346, 400, 402, 404, 406, 460)을 식별하고, 상기 제 1 메모리 블록으로 새 코드 섹션을 기록하도록 구성된 코드 섹션 컴팩터(code section compactor; 342)

를 포함하는 것을 특징으로 하는, 무선 통신 장치 상에서 새 코드 섹션을 다수의 현 코드 섹션으로 치밀화하기 위한 시스템.

**청구항 23**

제 22 항에 있어서, 상기 소프트웨어 업데이트(102)는 상기 새 코드 섹션과 명령 세트를 포함하는 것을 특징으로 하는, 무선 통신 장치 상에서 새 코드 섹션을 다수의 현 코드 섹션으로 치밀화하기 위한 시스템.

**청구항 24**

제 23 항에 있어서, 상기 새 코드 섹션은 부분 코드 섹션(partial code section)을 포함하는 것을 특징으로 하는, 무선 통신 장치 상에서 새 코드 섹션을 다수의 현 코드 섹션으로 치밀화하기 위한 시스템.

**청구항 25**

제 24 항에 있어서, 상기 부분 코드 섹션은 심벌 라이브러리를 포함하는 것을 특징으로 하는, 무선 통신 장치 상에서 새 코드 섹션을 다수의 현 코드 섹션으로 치밀화하기 위한 시스템.

**청구항 26**

제 22 항에 있어서, 다수의 코드 섹션 식별자와 이들의 해당 시작점 어드레스를 상기 비휘발성 메모리에 포함하는 코드 섹션 어드레스 테이블(code section address table; 326)로서, 이때, 상기 코드 섹션 컴팩터(342)가

상기 코드 섹션 어드레스 테이블(326)을 업데이트하도록 구성되는 상기 코드 섹션 어드레스 테이블(code section address table)

을 더 포함하는 것을 특징으로 하는, 무선 통신 장치 상에서 새 코드 섹션을 다수의 현 코드 섹션으로 치밀화하기 위한 시스템.

**청구항 27**

제 26 항에 있어서, 다수의 코드 심벌 식별자와 이들의 해당 코드 섹션 식별자와 메모리 오프셋 어드레스를 포함하는 심벌 오프셋 어드레스 테이블(symbol affset address table; 328)로서, 이때 상기 코드 섹션 컴팩터(342)는 상기 심벌 오프셋 어드레스 테이블을 업데이트하도록 구성되는 상기 심벌 오프셋 어드레스 테이블(symbol affset address table)

을 더 포함하는 것을 특징으로 하는, 무선 통신 장치 상에서 새 코드 섹션을 다수의 현 코드 섹션으로 치밀화하기 위한 시스템.

**청구항 28**

제 27 항에 있어서, 현 코드 섹션은 심벌 라이브러리와 미사용 섹션(340, 346)을 포함하며, 이때 상기 컴팩터(342)는 미사용 섹션으로 새 메모리 블록을 할당하고, 상기 새 메모리 블록에 새 코드 섹션을 기록하도록 더 구성되는 것을 특징으로 하는, 무선 통신 장치 상에서 새 코드 섹션을 다수의 현 코드 섹션으로 치밀화하기 위한 시스템.

**청구항 29**

제 28 항에 있어서, 다수의 미사용 섹션을 조합하고, 상기 조합된 미사용 섹션으로 새 메모리 블록을 할당하며, 상기 새 메모리 블록으로 새 코드 섹션을 기록하도록, 상기 컴팩터(342)가 더 구성되는 것을 특징으로 하는, 무선 통신 장치 상에서 새 코드 섹션을 다수의 현 코드 섹션으로 치밀화하기 위한 시스템.

**청구항 30**

제 28 항에 있어서, 현 코드 섹션을 일시적으로 저장하도록 구성되는 파일 시스템 섹션(110)을 더 포함하며, 이때 상기 컴팩터(342)는 비휘발성 메모리에 현 코드 섹션을 재배치하여 비휘발성 메모리의 연속적인 미사용 섹션을 생성하고, 상기 연속적인 미사용 섹션에 새 메모리 블록을 할당하며, 상기 새 메모리 블록에 상기 새 코드 섹션을 기록하도록 더 구성되는 것을 특징으로 하는, 무선 통신 장치 상에서 새 코드 섹션을 다수의 현 코드 섹션으로 치밀화하기 위한 시스템.

**청구항 31**

제 30 항에 있어서, 재배치된 현 코드 섹션에 대한 새 시작점 어드레스를 이용하여 코드 섹션 어드레스 테이블(326)을 업데이트하도록, 상기 컴팩터(342)가 더 구성되는 것을 특징으로 하는, 무선 통신 장치 상에서 새 코드 섹션을 다수의 현 코드 섹션으로 치밀화하기 위한 시스템.

**청구항 32**

제 31 항에 있어서, 현 코드 섹션의 재배치에 앞서서 무선 통신 장치의 동작을 중지하고, 상기 코드 섹션 어드레스 테이블(326)을 업데이트한 후에 상기 무선 통신 장치를 재설정하도록, 상기 컴팩터(342)가 더 구성되는 것을 특징으로 하는, 무선 통신 장치 상에서 새 코드 섹션을 다수의 현 코드 섹션으로 치밀화하기 위한 시스템.

**청구항 33**

제 22 항에 있어서, 비휘발성 메모리의 제 1 블록은 제 1 현 코드 섹션을 포함하며, 상기 제 1 현 코드 섹션이 상기 새 코드 섹션으로 대체되도록 상기 컴팩터(342)가 더 구성되는 것을 특징으로 하는, 무선 통신 장치 상에서 새 코드 섹션을 다수의 현 코드 섹션으로 치밀화하기 위한 시스템.

**청구항 34**

삭제

- 청구항 35  
삭제
- 청구항 36  
삭제
- 청구항 37  
삭제
- 청구항 38  
삭제
- 청구항 39  
삭제
- 청구항 40  
삭제
- 청구항 41  
삭제
- 청구항 42  
삭제
- 청구항 43  
삭제
- 청구항 44  
삭제
- 청구항 45  
삭제
- 청구항 46  
삭제
- 청구항 47  
삭제
- 청구항 48  
삭제
- 청구항 49  
삭제
- 청구항 50  
삭제

청구항 51

삭제

청구항 52

삭제

청구항 53

삭제

청구항 54

삭제

청구항 55

삭제

청구항 56

삭제

청구항 57

삭제

청구항 58

삭제

**명세서**

**기술분야**

<1> 본 발명은 무선 통신 장치에 관한 것으로서, 특히, 에어 링크 인터페이스를 통해 업데이트되는, 현장의 무선 통신 장치의 시스템 소프트웨어 내 코드 섹션들의 치밀화 시스템 및 방법에 관한 것이다.

**배경기술**

<2> 당 분야에 이미 존재하는 전화에 대한 소프트웨어 업데이트를 배포하는 것은 아주 혼란 일이다. 이 업데이트들은 전화가 제작되어 공공에게 배포된 후 소프트웨어적으로 발견되는 문제점에 관한 것들일 수 있다. 일부 업데이트는 전화에 새로운 특징의 이용을 포함할 수 있고, 서비스 제공자에 의해 제공되는 서비스의 이용을 포함할 수도 있다. 또다른 업데이트는 지역적 문제를 포함할 수 있고, 캐리어에 관련된 문제일 수도 있다. 예를 들어, 어떤 영역에서, 캐리어의 네트워크 배치는 핸드셋 상에 에어 링크 인터페이스 조건을 부과할 수 있어서, 부적절한 채널 검색, 부적절한 통화 종료, 부적절한 오디오, 등처럼 예기치않은 거동을 핸드셋이 나타낼 수 있다.

<3> 이러한 업데이트에 대한 전통적인 접근법은 무선 장치, 전화, 핸드셋 등으로 불리는 무선 통신 장치를 가장 가까운 캐리어 소매점/서비스센터에 리콜시키거나 이러한 변화 처리를 위해 제작자에게 리콜시키는 것이다. 이러한 업데이트에 관련된 비용은 비싸고 순이익을 잠식할 수 있다. 더욱이, 고객이 불편하고 짜증내기 쉽다. 가끔씩 실용적 해법은 고객에게 새 전화를 발급해주는 것이다.

<4> 무선 통신 장치 소프트웨어가 저렴하면서도 고객측면에서 편리하게 업그레이드될 수 있다면 바람직할 것이다.

<5> 상당한 시간동안 고객이 전화를 사용할 수 없는 불편함없이 무선 통신 장치 소프트웨어가 업그레이드될 수 있다면 바람직할 것이다.

<6> 무선 통신 장치 소프트웨어가 최소한의 기술 서비스 시간만으로, 서비스 센터에 통신 장치를 보낼 필요없이 업그레이드될 수 있다면 바람직할 것이다.

<7> 시스템 소프트웨어를 업데이트할 때 시스템 소프트웨어의 특정 코드 섹션만이 교체될 필요가 있도록, 무선 장

치 시스템 소프트웨어가 코드 섹션들로 구분될 수 있다면 바람직할 것이다. 이 코드 섹션들이 에어 링크를 통해 무선 장치에 전달될 수 있다면 또한 바람직할 것이다.

- <8> 시스템 소프트웨어가 현재 배열된 코드 섹션들보다 크기가 큰 코드 섹션들로 업데이트될 수 있다면 바람직할 것이다. 업데이트된 이러한 크기가 큰 코드 섹션들을 수용하도록 시스템 소프트웨어가 재배열될 수 있다면 또한 바람직할 것이다.
- <9> 특정 사용자들의 문제점을 해결하도록, 또는 필요에 부합하도록 이 지속성 데이터가 선택적으로 업데이트될 수 있다면 또한 바람직할 것이다.

**발명의 상세한 설명**

- <10> 무선 통신 장치 소프트웨어 업데이트는 가장 좋은 제품과 사용자 경험을 고객에게 제공한다. 사업상의 고비용 요인은 소프트웨어 업데이트를 위한 핸드셋 리콜을 포함한다. 이 업데이트는 사용자에게 추가적 서비스를 제공하기 위해 필요할 수도 있고, 제작 후 전화 사용시 발견되는 문제점을 해결하기 위해 필요할 수도 있다. 본 발명은, 업그레이드 코드 섹션이 대체될 코드 섹션보다 클 경우, 배포되어 현장에서 사용중인 핸드셋 소프트웨어를 에어 링크 인터페이스를 통해 실용적으로 업그레이드할 수 있게 한다.
- <11> 따라서, 무선 통신 장치 메모리에 저장된 소프트웨어 명령을 재조직하기 위한 방법이 제공된다. 이 방법은, 1) 연속적으로 어드레싱되는 다수의 제 2 메모리 블록을 생성함으로써, 각각의 메모리 블록을 해당 코드 섹션으로 식별함으로써, 그리고 식별된 메모리 블록에 코드 섹션들을 저장함으로써, 해당 시작 어드레스에서 코드 섹션이 시작하는, 다수의 현 코드 섹션에 무선 장치 시스템 소프트웨어를 저장하고, 2) 무선 통신 장치 에어 인터페이스를 통해 새 코드 섹션을 수신하며, 3) 업데이트를 위해 현 코드 섹션을 식별하고, 4) 코드 섹션 크기를 연산하며, 5) 코드 섹션 크기 연산에 따라 치밀화 스케줄을 발생시키고, 6) 현 코드 섹션의 크기를 조정하며, 7) 현 코드 섹션의 크기 조정에 따라 코드 섹션 시작 어드레스를 변경시키고, 8) 코드 섹션들을 파일 시스템 섹션으로 일시적으로 이동시키며, 9) 치밀화 스케줄에 따라 연속 어드레싱을 관리하도록 파일 시스템 섹션으로부터의 상기 코드 섹션들을 메모리 블록에게로 저장함으로써 식별된 현 코드 섹션을 새 코드 섹션으로 대체하고, 그리고 10) 업데이트된 시스템 소프트웨어를 실행하는 단계들을 포함한다.
- <12> 무선 통신 장치 시스템 소프트웨어의 소프트웨어 명령을 재조직하기 위한 상술한 방법 및 소프트웨어 명령의 재조직을 위한 무선 장치 시스템의 세부적 사항들이 아래에 제시된다.

**실시예**

- <24> 아래에 이어지는 실시예의 일부만이 과정, 단계, 로직 블록, 코드, 처리공정, 그리고 무선 장치 마이크로프로세서나 메모리 내 데이터 비트의 동작을 표현하는 다른 심벌을 들어 설명된다. 이 표현들은 데이터 처리 분야에 통상의 지식을 가진 자들에 의해 사용되는 수단이다. 과정, 마이크로프로세서 실행 단계, 데이터 아이템, 애플리케이션, 로직 블록, 처리공정 등은 요망 결과를 이끄는 단계들이나 명령들의 사리에 맞는 시퀀스로 간주된다. 이 단계들은 물리적 양의 물리적 조작을 필요로하는 단계들이다. 일반적으로, 반드시 그러한 것은 아니지만, 이 양들은 저장, 전송, 조합, 비교, 마이크로프로세서 기반 무선 장치의 조작이 가능한 전기/자기 신호의 형태를 취한다. 비트, 값, 요소, 심벌, 캐릭터, 데이터 아이템, 수치 등으로 이 신호들을 명명하는 것은 대중적으로 널리 사용되기 때문에 편리한 것으로 인식되고 있다. 메모리같은 물리적 소자들이 언급되는 데, 이들은 버스나 다른 전기적 연결을 통해 다른 물리적 소자들에 연결된다. 이 물리적 소자들은 로직 처리공정이나 애플리케이션과 상호작용할 수 있다고 간주될 수 있고, 따라서, 로직 동작에 "연결"된다. 예를 들어, 메모리는 추가적인 로직 동작에 코드를 저장허간 액세스할 수 있고, 애플리케이션은 실행을 위해 메모리로부터 코드 섹션을 호출할 수 있다.
- <25> 그러나, 이 용어들이 적절한 물리적 양에 관련된 것으로서, 이 양들에 적용된 편리한 라벨에 지나지 않음을 명심하여야 한다. 다음의 설명으로부터 명백하겠지만, 본 발명을 통해, "처리", "연결", "변환", "디스플레이", 또는 "프래프트", "결정", "디스플레이", "인식", "비교", "대체(교체)", "어드레스(해결처리)", "불러오기", 등같은 용어들을 이용한 설명은 컴퓨터 시스템의 레지스터와 메모리 내의 물리적 양으로 표현되는 데이터를 무선 장치 메모리나 레지스터, 또는 그 외 다른 이러한 정보 저장, 전송, 또는 디스플레이 장치 내의 물리적 양으로 표현되는 다른 데이터로 조작하고 변환하는 무선 장치 마이크로프로세서 시스템의 동작들을 의미한다.
- <26> 도 1은 전체 무선 장치 소프트웨어 관리 시스템(100)의 블록도표이다. 본 발명의 시스템 소프트웨어 조직은 아래에 상세하게 제시되며, 소프트웨어 관리 시스템(100)의 일반적 개관을 따른다. 일반 시스템(100)은 시스템 소

소프트웨어 업데이트 및 명령 세트(프로그램)의 전달 과정과, 무선 장치에서 전달된 소프트웨어의 설치 과정을 소개한다. 시스템 소프트웨어 업데이트와 패치 매니저 런타임 명령(PMRTI)(명령 세트나 동적 명령 세트로 알려짐)은 핸드셋 제작자에 의해 생성된다. 시스템 소프트웨어는 심벌 라이브러리로 조직된다. 심벌 라이브러리는 코드섹션들로 배열된다. 심벌 라이브러리가 업데이트되어야 할 때, 소프트웨어 업데이트(102)가 한개 이상의 코드 섹션으로 전달된다. 소프트웨어 업데이트는 이미 배포된 무선 장치에 송출되며, 그 무선 통신 장치(104)의 소프트웨어 업데이트(102)는 잘 알려진 기존의 에어, 데이터, 또는 메시지 전송 프로토콜을 이용하여 기지국(106)으로부터 별도의 통신들로 전송된다. 발명은 어떤 특정 전송 포맷에 국한되지 않는다. 왜냐하면, 무선 통신 장치가 시스템 소프트웨어 및 PMRTI 업데이트를 수신하기 위해 어떤 가용한 에어 전송 프로토콜을 처리하도록 용이하게 수정될 수 있기 때문이다.

<27> 이 시스템 소프트웨어는 여러 다른 서브시스템들의 집합체로 보일 수 있다. 코드 객체들은 이 추상적 서브시스템들 중 하나에 강력하게 연결될 수 있고, 결과적인 집합체가 심벌 라이브러리의 라벨을 달 수 있다. 이는 코드 베이스의 로직 파괴를 제공하고, 소프트웨어 패치와 처방(fixes)이 심벌 라이브러리 중 하나에 관련될 수 있다. 대부분의 경우에, 단일 업데이트가 한개나 두개의 심벌 라이브러리에 관련된다. 나머지 코드 베이스, 나머지 심벌 라이브러리는 불변으로 유지된다. 심벌 라이브러리의 표현은 코드와 상수를 취급하기 위한 메커니즘을 제공한다. 다른 한편, 판독-기록(Read-Write) 데이터는 모든 라이브러리에 대해 RAM 기반 데이터를 내포한 독자적이고 개별적인 판독-기록 라이브러리에 부합한다.

<28> 무선 장치(104)에 의해 수신되었을 때, 전송된 코드 섹션이 반드시 처리되어야 한다. 무선 장치는 비휘발성 메모리의 특정 코드 섹션을 덮어쓴다. 비휘발성 메모리(108)는 파일 시스템 섹션(FSS)(110)과 코드 저장 섹션(112)을 포함한다. 코드 섹션은 FSS(110)의 점유를 최소화시키기 위해 전송 전에 일반적으로 압축된다. 업데이트된 코드 섹션이 판독-기록 데이터를 동반하는 경우가 자주 있으며, 이는 각각의 심벌 라이브러리에 대해 모든 판독-기록 데이터를 내포한 또다른 종류의 심벌 라이브러리아. 시스템 소프트웨어가 실행 중일 때 휘발성 판독-기록 RAM(114)에 로딩되지만, 판독-기록 데이터는 비휘발성 메모리(108)에 저장될 필요가 항상 있으며, 따라서, 무선 장치가 리셋될 때마다 판독-기록 데이터가 휘발성 판독-기록 RAM(114)에 로딩될 수 있다. 이는 판독-기록 데이터가 휘발성 판독-기록 RAM으로 로딩되는 첫 번째를 포함한다. 아래에 상세하게 설명되겠지만, 판독-기록 데이터는 패치 매니저 코드 섹션으로 배열되는 것이 일반적이다.

<29> 시스템(100)은 가상 테이블의 개념을 포함한다. 이러한 테이블을 이용하여, 한 코드 섹션의 심벌 라이브러리들이 시스템 소프트웨어의 다른 부분(다른 코드 섹션)을 파괴(교체)하지 않으면서 패치(교체)될 수 있다. 가상 테이블은 효율성을 위해 휘발성 판독-기록 RAM(114)으로부터 실행된다. 코드 섹션 어드레스 테이블과 심벌 오프셋 어드레스 테이블은 가상 테이블이다.

<30> 업데이트된 코드 섹션들이 무선 장치(104)에 의해 수신되어 FSS(110)에 저장된다. 무선 장치 사용자 인터페이스(UI)는 통상적으로 새 소프트웨어가 가용함을 사용자에게 알릴 것이다. 사용자 인터페이스 프라프트에 따라 사용자는 이 통지를 확인하고 패치나 업데이트 동작을 신호한다. 대안으로, 업데이트 동작이 자동적으로 시행된다. 업데이트 처리가 실행됨에 따라 무선 장치가 표준 통신 작업을 수행하지 못할 수가 있다. 패치 매니저 코드 섹션은 휘발성 판독-기록 RAM(114)에 또한 로딩되는 비휘발성 판독-기록 드라이버 심벌 라이브러리를 포함한다. 비휘발성 판독-기록 드라이버 심벌 라이브러리는 코드 섹션들을 업데이트된 코드 섹션으로 덮어쓴다. 패치 매니저 코드 섹션은 판독-기록 데이터, 코드 섹션 어드레스 테이블, 그리고 심벌 오프셋 어드레스 테이블을 포함하고, 또한, 심벌 액세서 코드와 심벌 액세서 코드 어드레스도 포함한다. 업데이트된 코드 섹션들이 도입되면 이 데이터의 일부분들이 유효하지 않으며, 업데이트된 패치 매니저 코드 섹션들은 업데이트된 코드 섹션에 대해 유효한 판독-기록 데이터, 코드 섹션 어드레스 테이블, 그리고 심벌 오프셋 어드레스 테이블을 포함한다. 업데이트된 코드 섹션이 코드 저장 섹션(112)으로 로딩되면, 무선 장치가 리셋된다. 리셋 동작에 이어, 무선 장치는 업데이트된 시스템 소프트웨어를 실행할 수 있다. 패치 매니저 코드 섹션이 상술하지 않은 다른 심벌 라이브러리를 포함할 수도 있다. 이러한 다른 심벌 라이브러리들은 휘발성 판독-기록 메모리(114)로 로딩될 필요가 없다.

<31> 도 2는 에어 링크 인터페이스를 통해 명령 세트의 설치를 강조한, 소프트웨어 관리 시스템(100)의 블록도표이다. 시스템 소프트웨어 코드 섹션의 업데이트에 추가하여, 관리 시스템(100)은 패치 매니저 런타임 명령(PMRTI)이라 불리는 동적 명령 세트, 프로그램, 또는 패치 매니저 명령 세트(PMIS)를 다운로드하고 설치할 수 있다. PMRTI 코드 섹션(200)은 상술한 시스템 소프트웨어 코드 섹션과 같은 방식으로 무선 장치(104)에 전달된다. PMRTI 코드 섹션은 핸드셋에 컴파일된 명령으로 나타날 수 있는 이진(binary) 파일이다. PMRTI 코드 섹션은 조건부 실행 동작의 성능과 기본 수학적 동작의 성능을 제공할 만큼 충분히 포괄적이다. 예를 들어, RF 조정

PMRTI는 다음의 동작을 수행할 수 있다.

<32> IF RF CAL ITEM IS LESS THAN X

<33> EXECUTE INSTRUCTION

<34> ELSE

<35> EXECUTE INSTRUCTION

<36> PMRTI는 덧셈, 뺄셈, 곱셈, 나눗셈같은 기본적인 수학 연산을 지원할 수 있다. 시스템 소프트웨어 코드 섹션에 대하여, PMRTI 코드 섹션은 UI 프래프트에 따라 로딩될 수 있고, 무선 장치는 PMRTI가 코드 저장 섹션(112)으로 로딩된 후 리셋되어야 한다. PMRTI 코드 섹션이 어떤 가상 테이블이나 관독-기록 데이터에 연계될 경우, 업데이트된 패치 매니저 코드 섹션이 코드 저장 섹션(112)에서의 설치를 위해 PMRTI로 전송될 것이다. 대안으로, PMRTI가 유지되어 FSS(110)로부터 처리될 수 있다. 핸드셋(104)이 PMRTI 섹션의 모든 명령을 수행한 후, PMRTI 섹션이 FSS(110)로부터 삭제될 수 있다.

<37> PMRTI는 매우 강력한 런타임 명령 엔진이다. 핸드셋이 PMRTI 환경을 통해 전달되는 어떤 명령도 실행할 수 있다. 이 메커니즘은 RF 조정을 지원하는 데 사용될 수 있다. 보다 일반적으로, PMRTI는 제작자나 서비스 제공자에 의해 소프트웨어 문제가 인지되었을 때 무선 장치 소프트웨어를 원격 디버깅(remote debug)하는 데 사용될 수 있다. PMRTI는 데이터 분석, 디버그, 처방을 위해 새로이 다운로드받은 시스템 애플리케이션을 발견시킬 수 있다. PMRTI는 업데이트된 시스템 소프트웨어 코드 대신에 문제점에 대한 단기 처방과 분석에 대한 관독-기록 데이터 기반 업데이트를 제공할 수 있다. PMRTI는 무선 장치에 의한 이용을 위해 메모리 치밀화 알고리즘을 제공할 수 있다.

<38> 발명의 일부 태양에서, 시스템 소프트웨어를 심벌 라이브러리로 조직하는 것은 실행에 필요한 휘발성 메모리(114)와 비휘발성 메모리(108)의 크기에 영향을 미칠 수 있다. 이는 코드 섹션이 코드 섹션에 배열되는 심벌 라이브러리들보다 크기 때문이다. 이같이 큰 코드 섹션은 업데이트된 코드 섹션을 수용하기 위해 존재한다. 라이브러리의 집합체로 시스템 소프트웨어를 조직하는 것은 비휘발성 메모리 크기 요건에 영향을 미친다. 동일한 코드 크기에 대하여, 사용되는 비휘발성 메모리의 양은 코드 섹션의 크기가 그 안에 배열된 심벌 라이브러리보다 크기 때문에 높을 것이다.

<39> 소프트웨어 업데이트가 무선 장치에 전달되면, 소프트웨어 관리 시스템(100)은 메모리 치밀화를 지원한다. 메모리 치밀화는 데스크탑 컴퓨터의 디스크 디프래그먼트 애플리케이션(disk defragmentation application)과 유사하다. 치밀화 메커니즘은 메모리가 최적으로 사용되고 차후 코드 섹션 업데이트에 대해 우수한 균형을 유지함을 보장한다. 이때, 업데이트된 코드 섹션의 크기는 예측불가능하다. 시스템(100)은 패치(업데이트) 실행 중에 코드 저장 섹션을 분석한다. 시스템(100)은 교체되는 코드 섹션에 의해 점유되는 메모리 공간에 업데이트된 코드 섹션을 맞추려고 시도한다. 업데이트된 코드 섹션이 교체되는 코드 섹션보다 클 경우, 시스템(100)은 메모리(112)의 코드 섹션을 치밀화한다. 대안으로, 제작자나 서비스 제공자에 의해 치밀화가 연산될 수 있고, 치밀화 명령이 무선 장치(104)에 전달될 수 있다.

<40> 치밀화는 알고리즘의 복잡도와 방대한 분량의 데이터 이동 때문에 시간이 소요되는 공정일 수 있다. 치밀화 알고리즘은 어떤 공정을 시작하기 전에 가능성을 예측한다. UI 프래프트는 치밀화가 시도되기 전에 사용자로부터의 허가를 얻는 데 사용될 수 있다.

<41> 일부 발명의 태양에서, 모든 시스템 소프트웨어 코드 섹션들이 동시에 업데이트될 수 있다. 완전한 시스템 소프트웨어 업그레이드는 더 큰 FSS(110)를 필요로 할 것이다.

<42> 도 3은 무선 통신 장치의 소프트웨어 명령을 재조직하기 위한 발명에 따른 시스템을 도시하는 블록도표이다. 시스템(300)은 다수의 현 코드 섹션으로 구분되는 실행형 무선 장치 시스템 소프트웨어를 포함하는 메모리(108)의 코드 저장 섹션(112)을 포함한다. 코드 섹션 1(302), 코드 섹션 2(304), 코드 섹션 n(306), 그리고 패치 매니저 코드 섹션(308)이 도시된다. 그러나, 발명이 어떤 특정 수치의 코드 섹션에 제한되지는 않는다. 더욱이, 시스템(300)은 다수의 리소스 섹션들로 배열되는 다수의 제 1 심벌 라이브러리들을 추가로 포함한다. 심벌 라이브러리 1(310)이 코드 섹션 1(302)에, 심벌 라이브러리 2(312)와 3(314)이 코드 섹션 2(304)에 배열되며, 심벌 라이브러리 m(316)은 코드 섹션 n(306)에 배열된다. 각각의 라이브러리는 관련 기능을 가진 심벌들을 포함한다. 예를 들어, 심벌 라이브러리 1(310)은 무선 장치 LCD의 동작에 관련될 수 있다. 그후, 심벌은 디스플레이 기능에 관련될 수 있다. 아래 설명되는 바와 같이, 추가적인 심벌 라이브러리들이 패치 매니저 코드 섹션(308)에 배열된

다.

- <43> 도 4는 무선 장치 메모리의 블록도표이다. 도시되는 바와 같이, 메모리는 도 1의 코드 저장 섹션(112)이다. 메모리는 쓰기가능한 비휘발성 메모리이다(가령, 플래시 메모리). 코드 섹션이 반드시 FSS(110)와 같은 메모리에 저장될 필요는 없다. 본 발명의 시스템 소프트웨어 구조가 다수의 협력 메모리에 저장된 코드 섹션들로 구현될 수 있다. 코드 저장 섹션(112)은 다수의 제 2 연속 어드레싱 메모리 블록을 포함하며, 이때, 각각의 메모리 블록은 다수의 리소스 섹션들로부터 해당 코드 섹션을 저장한다. 따라서, 코드 섹션 1(302)이 제 1 메모리 블록(400)에 저장되고, 코드 섹션 2(304)가 제 2 메모리 블록(402)에 저장되며, 코드 섹션 n(306)이 제 n 메모리 블록(404)에 저장되고, 패치 매니저 코드 섹션(308)이 제 p 메모리 블록(406)에 저장된다.
- <44> 도 3과 4를 대조해보면, 각각의 코드 섹션의 시작점이 메모리의 해당 시작 어드레스에 저장되고, 심벌 라이브러리들이 코드 섹션의 시작점에서 시작되도록 배열된다. 즉, 각각의 심벌 라이브러리는 제 1 어드레스에서 시작하여 제 1 어드레스로부터 순서대로 어드레스 범위를 따라 진행된다. 예를 들어, 코드 섹션 1(302)은 코드 저장 섹션 메모리(112)의 제 1 시작 어드레스(408)("S"로 표시)에서 시작된다. 도 3에서, 심벌 라이브러리 1(310)은 제 1 코드 섹션의 시작점(318)에서 시작된다. 마찬가지로 코드 섹션 2(304)는 제 2 시작 어드레스(410)(도 4)에서 시작되고, 심벌 라이브러리 2는 코드 섹션 2의 시작점(320)(도 3)에서 시작된다. 코드 섹션 n(306)은 코드 저장 섹션 메모리(112)의 제 3 시작 어드레스(412)에서 시작되고(도 4), 심벌 라이브러리 m(316)은 코드 섹션 n(322)의 시점에서 시작된다(도 3). 패치 매니저 코드 섹션은 코드 저장 섹션 메모리(112)의 제 p 시작 어드레스(414)에서 시작하고, 패치 매니저 코드 섹션(310)의 제 1 심벌 라이브러리는 제 1 메모리 블록(400)에 궁극적으로 저장된다. 한 코드 섹션이 다수의 심벌 라이브러리를 포함할 경우(가령, 코드 섹션 2(304)), 다수의 심벌 라이브러리는 제 2 메모리 블록(402)의 경우에, 해당 메모리 블록에 저장된다.
- <45> 도 3에서, 시스템(300)은 패치 매니저 코드 섹션(308)에 배열되는 심벌 라이브러리에 포함된 심벌 종류로 코드 섹션 어드레스 테이블(326)을 추가로 포함한다. 코드 섹션 어드레스 테이블은 코드 섹션 식별자를 메모리의 해당 코드 섹션 시점 어드레스와 상호참조(cross-reference)한다.
- <46> 도 5는 도 3의 코드 섹션 어드레스 테이블(326)을 나타내는 테이블이다. 코드 섹션 어드레스 테이블(326)은 심벌 라이브러리에 대한 코드 섹션 시작 어드레스를 찾기 위해 참고된다. 예를 들어, 심벌 라이브러리 1의 한 심벌이 실행에 필요할 경우 시스템(300)이 코드 섹션 1을 찾는다. 코드 섹션 1의 시작 어드레스를 찾기 위해, 그래서 심벌 라이브러리 1의 심벌을 위치시키기 위해, 코드 섹션 어드레스 테이블(326)이 참고된다. 코드 섹션들에서 심벌 라이브러리들의 배열과, 코드 섹션을 테이블로 찾아가는 것은 코드 섹션을 이동시키거나 확장시킬 수 있다. 확장이나 이동 동작은 업그레이드된 코드 섹션을 설치하는 데 필요할 수 있다.
- <47> 도 3으로 돌아가서, 모든 심벌 라이브러리들이 반드시 코드 섹션의 시점에서 시작되는 것은 아니다. 도시되는 바와 같이, 심벌 라이브러리 3(314)은 코드 섹션 2(304)에 배열되지만, 코드 섹션 시작 어드레스(320)의 시작점은 아니다. 따라서, 심벌 라이브러리 3(314)의 한 심벌이 실행에 필요할 경우, 시스템(300)은 코드 섹션 2(304)의 시작 어드레스에 대한 코드 섹션 어드레스 테이블(326)을 참조한다. 아래 설명되는 바와 같이, 심벌 오프셋 어드레스 테이블이 심벌 라이브러리 3(314)의 심벌들을 위치시킨다. 심벌들이 동일한 코드 섹션으로 유지되기 때문에 심벌들이 다중 라이브러리 사이에 퍼지는 것은 문제가 되지 않는다.
- <48> 상술한 바와 같이, 각각의 심벌 라이브러리는 기능적으로 관련된 심벌들을 포함한다. 심벌은 루틴 바디(routine body), 변수, 또는 데이터 구조를 위치시키고 이용하기 위한 프로그래머-지정 명칭이다. 따라서, 심벌이 한개의 어드레스거나 한개의 값(value)일 수 있다. 심벌들은 내부적일 수도 있고 외부적일 수도 있다. 내부적 심벌은 현 코드 섹션의 범위를 넘으면 보이지 않는다. 보다 구체적으로, 내부적 심벌들은 다른 코드 섹션의 다른 심벌 라이브러리로 찾을 수 없다. 외부적 심벌은 코드 섹션 사이에서 사용되고 호출되며, 여러 다른 코드 섹션의 라이브러리로 찾을 수 있다. 심벌 오프셋 어드레스 테이블은 모든 외부적 심벌들의 리스트를 포함하는 것이 일반적이다.
- <49> 예를 들어, 심벌 라이브러리 1(310)은 무선 장치 디스플레이 상에 문자들을 발생시킨다. 이 라이브러리의 심벌들은 전화 번호, 성명, 시간, 또는 그 외 다른 디스플레이 특징들을 발생시킨다. 각각의 특징은 심벌이라고 불리는 루틴으로 발생된다. 가령, 심벌 라이브러리 1(310)의 한개의 심벌이 디스플레이 장치에 전화번호를 발생시킨다. 이 심벌은 X로 표시되며 외부적이다. 무선 장치가 전화통화를 수신하고 호출자 ID 서비스가 활성화되면, 시스템은 디스플레이 장치 상에 번호를 발생시키도록 "X" 심벌을 실행하여야 한다. 따라서 시스템은 X 심벌을 위치시켜야 한다.

- <50> 도 6은 심벌들로 구성된 도 3의 심벌 라이브러리 1(310)의 상세한 도면이다. 심벌들은 각각 코드 섹션 시작 어드레스로부터 이격되어 배치된다. 여러 상황에서, 심벌 라이브러리의 시작점은 코드 섹션의 시작점이지만, 이는 코드 섹션이 두개 이상의 심벌 라이브러리를 포함할 경우 사실이 아니다. 심벌 라이브러리 1(310)은 코드 섹션 1의 시작점에서 시작된다(도 3 참조). 도 6에 도시되는 바와 같이, "X"심벌은 심벌 라이브러리의 시작점으로부터 (03)만큼 이격되어 위치하고, "Y"심벌은 (15)만큼 이격되어 위치한다. 심벌 오프셋 어드레스들은 패치 매니저 코드 섹션의 심벌 오프셋 어드레스 테이블(328)에 저장된다(도 3 참조).
- <51> 도 7은 도 3의 심벌 오프셋 어드레스 테이블(328)을 도시하는 테이블이다. 심벌 오프셋 어드레스 테이블(328)은 심벌 식별자를 해당 오프셋 어드레스와, 메모리의 해당 코드 섹션 식별자와 상호참고(cross-reference)시킨다. 따라서, 시스템이 심벌 라이브러리 1의 "X"심벌을 실행하려할 경우, 심벌 오프셋 어드레스 테이블(328)이 코드 섹션에 대해 심벌의 정확한 어드레스를 위치시키기 위해 참고된다.
- <52> 도 3으로 되돌아가서, 다수의 제 1 심벌 라이브러리들은 이 심벌 라이브러리들의 실행시 설정되거나 참고되어야 하는 모든 판독-기록(read-write; RW) 데이터를 포함하는 것이 일반적이다. 예를 들어, 심벌 라이브러리가 조건부 문장에 따라 좌우되는 동작을 포함할 수 있다. 판독-기록 데이터 섹션은 조건부 문장(conditional statement)을 완성시키는 데 필요한 상태(status)를 결정하기 위해 참고된다. 본 발명은 모든 심벌 라이브러리로부터의 판독-기록 데이터를 공유 판독-기록 섹션으로 묶는다. 발명의 일부 태양에서, 판독-기록 데이터(330)는 패치 매니저 코드 섹션(308)에 배열된다. 대안으로(도시되지 않음), 판독-기록 데이터가 앞서와 다른 코드 섹션, 가령, 코드 섹션 n(306)으로 배열될 수 있다.
- <53> 다수의 제 1 심벌 라이브러리들은 찾는 심벌의 어드레스를 계산하기 위해 코드 섹션에 배열되는 심벌 액세스 코드를 또한 포함한다. 심벌 액세스 코드는 별도의 코드 섹션, 가령, 코드 섹션 2(304)의 어드레스에 배열되고 저장될 수 있다. 그러나 도시되는 바와 같이, 심벌 액세스 코드(332)는 패치 매니저 코드 섹션(308)의 어드레스에 배열되고 저장된다. 시스템(300)은 심벌 액세스 코드 어드레스의 저장을 위한 제 1 위치를 추가로 포함한다. 제 1 위치는 코드 저장 섹션(112)의 코드섹션일 수도 있고, 무선 장치의 별도 메모리 섹션의 코드 섹션일 수도 있다. 제 1 위치는 판독-기록 데이터와 동일한 코드 섹션에 배열될 수도 있다. 도시되는 바와 같이, 제 1 위치(334)는 판독-기록 데이터(330), 심벌 오프셋 어드레스 테이블(328), 코드 섹션 어드레스 테이블(326), 그리고 심벌 액세스 코드(332), 그리고 패치 라이브러리(패치 심벌 라이브러리)(336)로 패치 매니저 코드 섹션(308)에 저장된다.
- <54> 심벌 액세스 코드는 메모리에서 찾는 심벌의 어드레스를 연산하거나 발견하기 위해 코드 섹션 어드레스 테이블과 심벌 오프셋 어드레스 테이블에 액세스한다. 즉, 심벌 액세스 코드는 해당 심벌 식별자와 해당 코드 섹션 식별자를 이용하여 찾는 심벌의 어드레스를 연산한다. 예를 들어, 심벌 라이브러리 1의 "X"심벌을 찾을 경우, "X"심벌에 해당하는 심벌 식별자(심벌 ID) "X\_1"을 찾기 위해 심벌 액세스가 호출된다(도 7 참조). 심벌 액세스 코드는 "X\_1"심벌 식별자가 코드 섹션 1의 시작점으로부터 (03)만큼 이격되는 지를 결정하기 위해 심벌 오프셋 어드레스 테이블을 참고한다. 심벌 액세스 코드는 코드 섹션 식별자(코드 섹션 ID) "CS\_1"에 관련된 시작 어드레스를 결정하기 위해 코드 섹션 어드레스 테이블을 참고한다. 이 방식으로, 심벌 식별자 "X\_1"이 (00100)의 어드레스로부터 (03)만큼 이격되거나 (00103)에 위치함을 심벌 액세스 코드가 결정한다.
- <55> 심벌 "X"는 실제 코드의 일부분이기 때문에 예약된 명칭이다. 다시 말해서, 이와 관련된 절대적 데이터를 가진다. 이 데이터는 하나의 어드레스일 수도 있고 하나의 값(value)일 수도 있다. 심벌 식별자는 심벌 추적을 위해 생성되는 별칭이다. 심벌 오프셋 어드레스 테이블과 코드 섹션 어드레스 테이블은 예약된 심벌과 코드 섹션 명칭과의 혼동을 피하기 위해 식별자와 함께 기능한다. 동일한 심벌 명칭이 여러 심벌 라이브러리 사이에서 사용되는 것도 또한 가능하다. 식별자 사용은 이 심벌들간 혼동을 방지한다.
- <56> 도 1로 되돌아가서, 시스템(300)은 판독-기록 휘발성 메모리(114), 통상적으로 RAM을 추가로 포함한다. 판독-기록 데이터(330), 코드 섹션 어드레스 테이블(326), 심벌 오프셋 어드레스 테이블(328), 심벌 액세스 코드(332), 그리고 심벌 액세스 코드 어드레스(334)가 시스템 소프트웨어 실행 중 액세스를 위한 패치 매니저 코드 섹션으로부터 판독-기록 휘발성 메모리(114)로 로딩된다.
- <57> 당 분야에 잘 알려진 바와 같이, RAM 에 저장되는 코드에 대한 액세스 시간은 플래시같은 비휘발성 메모리에 대한 액세스보다 상당히 작다.
- <58> 도 3으로 되돌아가서, 메모리 블록이 그 안에 저장되는 해당 코드 섹션을 정확하게 수용할 수 있는 크기를 가지지만, 심벌 라이브러리들이 배열될 코드 섹션들을 반드시 채울 필요는 없다. 달리 말하자면, 다수의 리소스 섹

션들 각각은 배열되는 심벌 라이브러리들을 수용하는 바이트 크기를 가지며, 연속적으로 어드레싱되는 메모리 블록들 각각은 해당 코드 섹션들을 수용하는 바이트 크기를 가진다. 가령, 코드 섹션 1(302)은 100 바이트의 길이를 가진 심벌 라이브러리를 수용하기 위해 100 바이트 섹션일 수 있다. 제 1 메모리 블록은 코드 섹션 1의 바이트 크기와 일치하기 위해 100바이트일 것이다. 그러나, 코드 섹션 1에 로딩되는 심벌 라이브러리가 100바이트보다 작을 수 있다. 도 3에 도시되는 바와 같이, 심벌 라이브러리 1(310)이 100바이트보다 작기 때문에 코드 섹션 1(302)은 사용하지 않은 섹션(340)을 가진다. 따라서, 다수의 리소스 섹션들 각각은 배열된 심벌 라이브러리들을 수용하는 데 필요한 크기보다 더 큰 크기를 가질 수 있다. 코드 섹션 크기를 과설정(oversizing)으로서, 업데이트된 더 큰 심벌 라이브러리들을 수용할 수 있다.

<59> 연속적으로 어드레싱되는 메모리 블록들은 물리적 메모리 공간을 가변 크기의 로직 블록으로 분할하는 것에 관련된다. 코드 섹션과 메모리 블록들은 코드 섹션이 메모리에 저장될 때 실질적으로 교환가능한 항목들이다. 코드 섹션의 개념은 심벌 라이브러리보다 큰 코드 섹션을 식별하는 데 사용되고, 이동하거나 조작될 때 코드 섹션의 심벌 라이브러리들의 집합체를 식별하는 데 사용된다.

<60> 도 3에 도시되는 바와 같이, 시스템(300)은 패치 라이브러리(336)라 불리는 패치 심벌 라이브러리를 포함하여, 코드 저장 섹션의 새 코드 섹션을 현 코드 섹션과 함께 배열한다. 코드 저장 섹션에서 새 코드 섹션을 현 코드 섹션과 함께 배열하는 것은 업데이트된 실행형 시스템 소프트웨어를 형성한다. 패치 매니저(336)는 새 코드 섹션을 현 코드 섹션과 함께 배열할 뿐 아니라, 코드 섹션들을 업데이트된 코드 섹션으로 대체한다.

<61> 도 4를 참고해보면, 메모리(108)의 파일 시스템 섹션(110)이 새 코드 섹션(450)과 업데이트된 패치 매니저 코드 섹션(452)처럼 새 코드 섹션을 수신한다. 파일 시스템 섹션은 현 코드 섹션과 함께 새 코드 섹션을 배열하기 위한 명령을 포함하는 제 1 패치 매니저 런타임 명령(PMRTI)(454)을 또한 수신한다. 도 1에 도시되는 바와 같이, 에어 링크 인터페이스(150)는 새 코드 섹션이나 업데이트된 코드 섹션을 수신하고 또한 제 1 PMRTI를 수신한다. 에어 링크 인터페이스(150)가 안테나로 나타나지만, 에어 링크 인터페이스가 RF 송수신기, 기지대역 회로, 그리고 복조 회로(도시되지 않음)를 포함한다는 것을 이해하여야 할 것이다. 파일 시스템 섹션(110)은 에어 링크 인터페이스(150)를 통해 수신한 새 코드 섹션을 저장한다. 관독-기록 휘발성 메모리(114)로부터 실행되는 패치 라이브러리(336)는 코드 저장 섹션의 제 1 코드 섹션, 가령, 코드 섹션 n(306)을, 제 1 PMRTI(454)에 따라 새(또는 업데이트된) 코드 섹션으로 대체한다. 통상적으로, 패치 매니저 코드 섹션(308)은 업데이트된 패치 매니저 코드 섹션(452)으로 대체된다. 코드 섹션들이 대체되면, 패치 라이브러리(336)는 코드 저장 섹션(112)의 제 1 코드 섹션, 가령, 코드 섹션 n(306)을 파일 시스템 섹션(110)의 업데이트된 코드 섹션, 가령, 코드 섹션(450)으로 덮어쓴다. 극단적인 경우에, 코드 저장 섹션(112)의 모든 코드 섹션들이 업데이트된 코드 섹션들로 대체된다. 즉, FSS(110)는 다수의 제 2 업데이트 코드 섹션을 수신하며, 패치 라이브러리(336)는 코드 저장 섹션(112)의 다수의 리소스 섹션을 다수의 제 2 업데이트 코드 섹션으로 대체한다. 물론, FSS(110)는 에어 링크 인터페이스를 통해 수신한 다수의 제 2 업데이트 코드 섹션을 수용할만큼 충분히 크다.

<62> 상술한 바와 같이, 업데이트된 코드 섹션들은 관독-기록 데이터 코드 섹션, 코드 섹션 어드레스 테이블 코드 섹션, 심벌 라이브러리, 심벌 오프셋 어드레스 테이블 코드 섹션, 심벌 액세스서 코드 섹션, 또는 새 패치 라이브러리를 가진 코드 섹션을 포함할 수 있다. 이러한 모든 코드 섹션들은 관련 심벌 라이브러리 및 심벌을 가지면서, 구분된 독립적 코드 섹션으로 저장될 수 있다. 이 코드 섹션들 각각은 독자적인 업데이트 코드 섹션으로 대체될 것이다. 즉, 업데이트된 관독-기록 코드 섹션이 수신되어, 코드 저장 섹션의 관독-기록 코드 섹션을 대체할 것이다. 업데이트 코드 섹션 어드레스 테이블 코드 섹션이 수신될 것이고 코드 저장 섹션의 코드 섹션 어드레스 테이블 코드 섹션을 대체할 것이다. 업데이트 심벌 오프셋 어드레스 테이블 코드 섹션이 수신될 것이고 코드 저장 섹션의 심벌 오프셋 어드레스 테이블 코드 섹션을 대체할 것이다. 업데이트된 심벌 액세스서 코드 섹션이 수신될 것이고 코드 저장 섹션의 심벌 액세스서 코드 섹션을 대체할 것이다. 마찬가지로, 업데이트된 패치 매니저 코드 섹션이 수신될 것이고 코드 저장 섹션의 패치 매니저 코드 섹션을 대체할 것이다.

<63> 그러나, 상술한 코드 섹션들이 패치 매니저 코드 섹션에서 함께 묶이는(bundled) 것이 통상적이다. 따라서 패치 매니저 코드 섹션(308)이 업데이트된 패치 매니저 코드 섹션(450)으로 대체될 때, 코드 저장 섹션의 관독-기록 코드 섹션이 시스템 섹션(110)으로부터 업데이트된 관독-기록 코드 섹션으로 대체된다. 마찬가지로, 업데이트된 패치 매니저 코드 섹션(450)이 설치될 때 코드 섹션 어드레스 테이블, 심벌 오프셋 어드레스 테이블, 심벌 액세스서 코드 섹션, 그리고 패치 라이브러리가 대체된다. 새 관독-기록 새 코드 섹션 어드레스 테이블, 새 심벌 오프셋 어드레스 테이블, 새 심벌 액세스서 코드, 그리고 새 패치 라이브러리를 업데이트된 패치 매니저 코드 섹션(450)으로, 코드 저장 섹션의 현 코드 섹션과 함께 배열하는 것은 업데이트된 실행형 시스템 소프트웨어를 형성

한다.

- <64> 파일 시스템 섹션(110)이 업데이트된 심벌 액세스 코드 어드레스를 수신할 때, 패치 매니저는 메모리 내 제 1 위치의 심벌 액세스서 코드 어드레스를 업데이트된 심벌 액세스서 코드 어드레스로 대체한다. 상술한 바와 같이, 메모리(334) 내 제 1 위치는 패치 매니저 코드 섹션 내에 놓인다(도 3 참조).
- <65> 도 3에 도시되는 바와 같이, 패치 라이브러리(308)는 컴팩터(compactator), 또는 컴팩터 심벌 라이브러리(342)를 또한 포함한다. 컴팩터(342)는 구분된 독립적 코드 섹션으로 구현될 수도 있지만, 시스템 소프트웨어 업그레이드에 관련된 기능들을 단일 패치 매니저 코드 섹션으로 묶는 것이 효율적이고 유용하다. 일반적으로, 컴팩터(342)는 코드 섹션들의 크기를 재설정한다고 말하여지며, 따라서 새 섹션들이 코드 저장 섹션(112)의 현 코드 섹션과 함께 배열될 수 있다.
- <66> 도 3 및 4와는 달리, 파일 시스템 섹션(110)이 에어 링크 인터페이스(150)를 통한 업데이트를 위해 현 코드 섹션을 식별하기 위한 명령으로 치밀화 명령 세트를 수신한다(도 1 참조). 예를 들어, 치밀화 명령 세트는 제 1 PMRTI(454)일 수 있으며, 이 치밀화 명령이 다른 업데이트 명령과 반드시 한 세트로 묶일 필요는 없다. 컴팩터(342)는 코드 저장 섹션의 식별된 현 코드 섹션을 새 코드 섹션으로 대체한다. 컴팩터(342)가 상술한 패치 라이브러리(336)와 협조하여 동작함을 이해하여야 한다. 달리 말하자면, 컴팩터(342)는, 시스템 소프트웨어 업데이트 처리에 코드 섹션 크기 재설정이 필요한 상황에서, 패치 라이브러리(336)를 돕도록 호출된다.
- <67> 컴팩터는 크기조정된 해당 코드 섹션이 코드 저장 섹션에 저장되도록 메모리 블록의 크기를 조정한다(도 4 참조). 예를 들어, 파일 시스템 섹션(110)이 제 1 크기의 새 코드 섹션(450)을 수신한다. 동시에, 또는 다른 메시지에, 무선 장치가 동적 명령 세트, 즉, PMRTI(454)와, 업데이트된 패치 매니저 코드 섹션(452)을 수신한다. 치밀화 명령 세트(454)는 상기 제 1 크기보다 작은 제 2 크기의 현 코드 섹션, 즉, 코드 섹션2(304)를 식별한다. 업데이트된 코드 섹션(450)의 크기가 대체되는 코드 섹션보다 크기 때문에, 치밀화 동작이 반드시 실행되어야 한다. 컴팩터(342)는 식별된 현 코드 섹션(304)에 관련된 제 2 메모리 블록(402)의 크기를 적어도 제 1 크기 이상으로 증가시킨다. 컴팩터는 (패치 라이브러리(336)와 협조하여) 해당 메모리 블록(402)에 저장된 식별된 현 코드 섹션(304)을 새 코드 섹션(450)으로 대체한다.
- <68> 상술한 바와 같이, 코드 섹션들이 그 안에 배열된 심벌 라이브러리에 대해 더 큰 크기를 가지는 것은 흔한 일이다. 컴팩터(342)는 해당 코드 섹션 내에 배열되는 심벌 라이브러리들의 크기를 결정하고, 코드 섹션들 내에 배열되는 심벌 라이브러리 크기에 보다 가깝게 부합하도록 코드 섹션 크기를 조정한다. 일반적으로, 컴팩터(342)는 차후의 코드 섹션 크기조정 및 업데이트 동작을 위해 코드 섹션을 최적으로 크기조정한다.
- <69> 상술한 예에서, 컴팩터(342)는 제 2 메모리 블록(402)에 인접한 코드 섹션들로부터 비사용 영역을 취함으로써 코드섹션을 최적으로 크기조정시킬 수 있다. 도시되는 바와 같이, 코드 섹션 1(302)과 코드 섹션 3(344)이 각각 비사용 영역(340, 346)을 가진다. 컴팩터는 제 1 메모리 블록(302)과 제 3 메모리 블록(460)의 크기 조정을 선택할 수 있고, 비사용 영역(340, 346)을 모두 취하여, 새 코드 섹션(450)용으로 충분히 큰 제 2 메모리 블록(402)을 생성할 수 있다.
- <70> 도 8은 일례의 치밀화 동작의 결과를 보여주는 블록도표이다. 코드 섹션 2는 더 큰 새(업데이트된) 코드 섹션(450)으로 대체되었다. 코드 섹션 1(302)의 비사용 섹션(340)이 코드 섹션 3(344)의 비사용 영역(360)처럼 감소되었다. 각각의 코드 섹션에 일부 비사용 영역을 남겨둠으로서, 효율적인 차후 확장 및 치밀화 동작을 기대할 수 있다. 예를 들어, 모든 비사용 영역(360)이 현 업데이트 동작에서 업데이트된 코드 섹션(450)을 수용하도록 사용될 경우, 차후 코드 섹션 3(344)의 확장은 인접 코드 섹션에서 치밀화 동작을 필요로하게 될 것이다. 본 예가 두 인접 메모리 블록의 크기조정만을 설명하지만, 일부 치밀화 동작은 일부 또는 모든 코드 섹션들이 새 코드 섹션 업데이트를 수용하도록 크기조정되어야 함을 요구할 수 있다.
- <71> 컴팩터(342)는 코드 섹션 어드레스 테이블로부터 시작 어드레스에 액세스하여 코드 섹션 크기를 측정하고, 심벌 오프셋 어드레스 테이블로부터 심벌 오프셋 어드레스에 액세스하여 해당 코드 섹션 내에 배열된 심벌 라이브러리들의 크기를 측정한다. 또한, 발명의 일부 태양에 따르면, 패치 매니저 코드 섹션(308)에 반드시 위치해야만 하는 것이 아닌 다른 심벌 라이브러리들에 컴팩터가 의존하고, 작업 실행을 위해 컴팩터가 이와 협조한다. 그러나, 컴팩터가 코드 섹션 어드레스 테이블과 심벌 오프셋 어드레스 테이블에 액세스하기 위해 상술한 심벌 액세스서 코드와 함께 동작하는 것이 바람직하다.
- <72> 코드 섹션 어드레스 테이블이 각각의 코드 섹션의 시작 어드레스를 포함하기 때문에, 인접 코드 섹션들의 시작 어드레스를 이용한 간단한 산술 연산이 코드 섹션의 크기를 결정할 수 있다. 마찬가지로, 심벌 라이브러리 내

최종 심벌의 어드레스가 코드 섹션 시작 어드레스와 함께 사용될 수 있어서, 해당 코드 섹션 내 배열되는 심벌 라이브러리들의 대략적 크기를 결정할 수 있고 따라서 비사용 영역의 크기를 결정할 수 있다. 예를 들어, 코드 섹션 1(302)의 시작 어드레스가 (0100), 코드 섹션 2(304)의 시작 어드레스가 (0200), 그리고 심벌 라이브러리 (310)의 최종 심벌 어드레스가 (0170)일 경우, 비사용 영역(340)의 크기는 대략 (0200-0170)이다.

<73> 그러나, 여러 심벌들이 여러 연속 어드레스에 대해 분포되고, 최종 심벌의 오프셋 어드레스가 심벌 라이브러리가 종료되는 어드레스가 반드시 같지는 않다. 위 예에서, 최종 심벌이 (0170)에서 시작되어 (0173)에서 종료될 경우, 비사용 공간(340)의 크기는 실제 (0200-0173)이다. 발명의 일부 태양에 따르면, 코드 저장 섹션(112)이 심벌 라이브러리의 종점을 유효화하기 위해 종료 심벌을 가진 심벌 라이브러리들을 포함한다. 예를 들어 도 6으로 돌아가서, 심벌 "Z"는 1바이트 종료 심벌이다. 즉, 심벌 "Z"의 어드레스가 심벌 라이브러리 1의 종점을 유효화한다. 비교기(342)는 종료 심벌 오프셋 어드레스를 이용하여, 심벌 액세스시 코드의 동작을 통해, 해당 코드 섹션 내에 배열되는 심벌 라이브러리들의 크기를 측정할 수 있다.

<74> 대안으로, 코드 저장 섹션(112)은 심벌 라이브러리들의 크기를 유효화하기 위해 크기 심벌을 가진 심벌 라이브러리들을 포함한다. 예를 들어, 심벌 "Y"(도 6)는 심벌 라이브러리 1의 크기를 유효화시키는 수치 (0073)일 수 있다. 컴팩터(342)는 해당 코드 섹션 내에 배열되는 심벌 라이브러리들의 크기를 측정하기 위해 크기 심벌에 액세스한다. 발명의 다른 태양에 따르면, 모든 심벌 라이브러리들에 대한 크기 심벌들이 패치 매니저 코드 섹션의 한 코드 섹션에 테이블로 저장된다.

<75> 컴팩터(342)는 파일 시스템 섹션(110)의 새 코드 섹션(450) 크기를 측정하여, 새 코드 섹션이 코드 저장 섹션(112) 내 현 코드 섹션들과 함께 배열될 수 있는 지를 결정한다. 컴팩터(342)는 해당 코드 섹션 내에 배열되는 심벌 라이브러리들의 크기를 측정함에 따라, 그리고 새 코드 섹션의 크기를 측정함에 따라 이 결정을 행한다. 치밀화가 새 코드 섹션을 성공적으로 설치하지 못할 경우, 컴팩터(342)는 자동적으로 동작을 어보트한다. 대안으로, 컴팩터가 성공 가능성을 연산하고, 무선 장치 사용자에게 그 가능성을 디스플레이하며, 그리고 치밀화 동작의 계속 여부를 사용자가 결정할 수 있게 한다. 성공 가능성에 상관없이, 컴팩터(342)가 치밀화 및 업데이트 처리과정 중 시스템 소프트웨어의 실행을 중지시켜야 하기 때문에, 무선 전화는 사용자에게 치밀화 과정의 시작 옵션을 사용자에게 제공한다.

<76> 업데이트된 코드 섹션의 크기를 결정하기 위해, 컴팩터(342)는 치밀화 명령 세트 데이터에 의존한다. 파일 시스템 섹션(110)은 새 코드 섹션(450)의 크기를 포함하는 치밀화 명령 세트(454)를 수신한다. 컴팩터(342)는 이 치밀화 명령 세트(454)에 액세스하여, 파일 시스템 섹션(110) 내 새 코드 섹션(450)의 크기를 결정하고, 해당 코드 섹션 내 배열되는 심벌 라이브러리들의 크기 측정 및 새 코드 섹션 크기 결정에 따라 새 코드 섹션이 코드 저장 섹션(112) 내 현 코드 섹션들과 함께 배열될 수 있는 지를 결정한다.

<77> 코드 저장 섹션의 비사용 메모리 블록 섹션이 클 경우 치밀화 동작을 피할 수 있다. 컴팩터(342)는 코드 저장 섹션 내 비사용 메모리 블록의 크기를 결정하여, 비사용 메모리 블록의 크기가 새 코드 섹션 크기보다 크거나 같을 경우 비사용 메모리 블록에 새 코드 섹션을 저장한다. 코드 섹션들의 크기가 조정되고 식별된 코드 섹션들이 업데이트된 코드 섹션들로 대체되면, 컴팩터(342)는 코드 저장 섹션에 저장된 코드 섹션들의 시작 어드레스를 변경시킨다. 상기 예에서, 크기조정 동작이 (0200) 대신 (0175)의 시작 어드레스에서 새 코드 섹션(450)을 시작시킴으로서 구현될 경우, 코드 섹션 어드레스 테이블이 새 시작 어드레스를 반영하도록 변경되어야 한다.

<78> 도 9a-9e는 일례의 치밀화 스케줄의 동작을 설명하는 블록도표이다. 휘발성 메모리(114)로부터 동작하는 컴팩터(342)가 성공적 치밀화의 가능성을 연산하고 코드 섹션 크기를 연산한 후, 컴팩터(342)가 치밀화 스케줄을 발생시킨다. 치밀화 스케줄은 코드 섹션들이 이동하는 순서를 포함하고, FSS(110) 내 임시 위치를 포함한다. 파일 시스템 섹션(110)은 코드 저장 섹션(112)으로부터 코드 섹션들을 일시적으로 저장하고, 컴팩터(342)는 치밀화 스케줄에 따라, 연속적 어드레싱을 관리하도록 파일 시스템 섹션(110)으로부터의 코드 섹션을 코드 저장 섹션(112) 메모리 블록에 저장한다.

<79> 계속하여 상기 예를 볼 때 도 9a에서, 컴팩터(342)는 업데이트된 코드 섹션과 새 코드 섹션(450)의 크기를 결정하기 위해 치밀화 명령 세트(제 1 PMRTI)(454)를 참조한다. 도 9b에서, 컴팩터(342)는, 코드 섹션의 성공 가능성과 크기를 연산한 후, 제 3 메모리 블록(460)으로부터 파일 시스템 섹션(110)에게로 코드 섹션 3(344)을 이동시킴으로서 처리과정을 시작한다.

<80> 도 9c에서, 새 코드 섹션(450)이 FSS(110)로부터 제 2 메모리 블록(402)으로 이동한다. 코드 섹션 1(302)의 크기가 조정되면, 컴팩터(342)는 새 코드 섹션의 시작 어드레스가 비사용 영역(340)에서 시작하게 한다(도 3

참조). 상술한 바와 같이, 제 2 메모리 블록의 시작점이 (0200)에서 (0175)로 변경된다. 즉, 코드 섹션 1(302)이 치밀화되었다. 도 9d에서, 코드 섹션 3(344)이 FSS(110)로부터 제 3 메모리 블록(460)으로 이동되었다. 이 이동은 코드 섹션 3(344)을 새로운 시작 어드레스에서 시작하게 한다. 코드 섹션을 새 시작 어드레스에서 시작하는 것은 코드 섹션의 치밀화를 포함할 수 있다. 즉, 비사용 영역(346)의 일부가 사용될 수 있다.

- <81> 도 9e에서, 패치 매니저 코드 섹션(308)이 새(업데이트된) 패치 매니저 코드 섹션(452)으로 대체된다. 통상적으로, 패치 매니저는 코드 섹션 어드레스 테이블 코드 심벌 라이브러리를 포함한다(도 3 참조). 그러나, 치밀화 결과가 반드시 사전에 알려진 것이 아니기 때문에 업데이트된 코드 섹션 어드레스 테이블이 반드시 정확한 것은 아니다. 그후, 컴팩터(342)는 메모리 블록에 저장된 코드 섹션들의 시작 어드레스를 변경함에 따라, 업데이트된 코드 섹션 어드레스 테이블을 발생시킨다. 컴팩터(342)는 코드 저장 섹션(112)에 저장된 업데이트된 코드 섹션 어드레스 테이블을, 발생된 업데이트된 코드 섹션 어드레스 테이블로 덮어쓴다. 그후, 새 코드 섹션이 현 코드 섹션과 함께 배열된 후 업데이트된 시스템 소프트웨어가 업데이트된 코드 섹션 어드레스 테이블에 액세스한다.
- <82> 그러나, 업데이트된 코드 섹션과 현 코드 섹션에 대한 심벌 오프셋 어드레스가 사전에 연산될 수 있기 때문에 심벌 오프셋 어드레스 테이블이 달리 취급된다. 파일 시스템 섹션(110)은 업데이트된 심벌 오프셋 어드레스 테이블로 새 코드 섹션, 통상적으로 패치 매니저 코드 섹션(452)을 수신하며, 업데이트된 시스템 소프트웨어는 새 코드 섹션(452)이 현 코드 섹션과 함께 배열된 후 업데이트된 심벌 오프셋 어드레스 테이블에 액세스한다.
- <83> 발명에 대한 상술한 설명 중 많은 부분이 코드 섹션 크기 및 치밀화 스케줄 연산에 관한 컴팩터의 동작에 대하여 이루어졌다. 그러나, 발명의 일부 태양에 따르면 컴팩터(342)가 이 연산들의 책임으로부터 구제된다. 발명의 이러한 태양에 따르면, 파일 시스템 섹션(110)이 코드 섹션 크기조정 명령 및 치밀화스케줄을 포함하는 치밀화 명령(454)을 수신한다. 그후, 컴팩터(342)는 코드 섹션 크기조정 명령에 따라 코드 섹션들의 크기를 조정한다. 앞서와 같이, 파일 시스템 섹션(110)은 코드 저장 섹션(112)으로부터 코드 섹션들을 일시적으로 저장하고, 컴팩터(342)는 치밀화 스케줄에 따라, 연속적 어드레스를 관리하기 위해 파일 시스템 섹션으로부터(110) 코드 섹션들을 코드 저장 섹션(112) 메모리 블록에 저장한다.
- <84> 파일 시스템 섹션(110)은 업데이트된 코드 섹션 어드레스 테이블과 업데이트된 심벌 오프셋 어드레스 테이블로, 가령, 업데이트된 패치 매니저 코드 섹션(452)같은 새 코드 섹션을 또한 수신한다. 코드 섹션 크기 조정 및 치밀화 스케줄이 사전에 계산되었기 때문에, 이 업데이트된 테이블들은 새 코드 섹션 시작 어드레스를 정확하게 반영한다. 새 코드 섹션이 현 코드 섹션과 함께 배열된 후, 업데이트된 시스템 소프트웨어는 업데이트된 심벌 오프셋 어드레스 테이블과 업데이트된 코드 섹션 어드레스 테이블에 액세스한다.
- <85> 도 1에 도시되는 바와 같이, 휘발성 메모리(114)는 시스템 소프트웨어 실행을 위해 코드 저장 섹션(112)으로부터 로딩되는 제 1, 2 테이블 코드 섹션(코드 섹션 어드레스 테이블 및 심벌 오프셋 어드레스 테이블)을 포함한다. 패치 매니저 코드 섹션(308)으로부터 다른 중요한 라이브러리들도 휘발성 메모리(114)로 로딩된다. 새 코드 섹션들이 코드 저장 섹션(112)의 현 코드 섹션들과 함께 배열되어, 무선 통신 장치 리셋에 따라 업데이트된 시스템 소프트웨어를 형성한다. 리셋 시에, 업데이트된 코드 섹션 어드레스 테이블과 업데이트된 심벌 오프셋 어드레스 테이블이 휘발성 메모리(114)로 로딩되고, 시스템 소프트웨어가 그후 업데이트된 테이블로 실행된다.
- <86> 도 10a-b는 무선 통신 장치 메모리에 저장된 소프트웨어 명령을 재조직하기 위한 본 발명의 방법을 설명하는 순서도이다. 이 방법은 단계 1000에서 시작한다. 단계 1001a는 시스템 소프트웨어를 다수의 제 1 심벌 라이브러리로 형성하며, 이때, 각각의 심벌 라이브러리는 한개 이상의 심벌들로 구성된다. 단계 1001b는 다수의 제 1 심벌 라이브러리들을 다수의 제 2 코드 섹션들로 배열한다. 단계 1002는 다수의 현 코드 섹션에 무선 장치 시스템 소프트웨어를 저장한다. 단계 1004는 새 코드 섹션을 수신한다. 새 코드 섹션을 수신하는 것(단계 1004)은 무선 통신 장치 에어 인터페이스를 통해 새 코드 섹션을 수신하는 과정을 포함한다.
- <87> 단계 1006은 현 코드 섹션의 크기를 조정한다. 통상적으로, 현 코드 섹션의 크기 조정은 시스템 소프트웨어의 동작을 중단하는 과정을 포함한다. 단계 1008은 무선 장치용 업데이트된 시스템 소프트웨어를 형성하기 위해 현 코드 섹션과 함께 새 코드 섹션을 배열한다. 단계 1010은 업데이트된 시스템 소프트웨어를 실행한다. 통상적으로, 단계 1005a는 업데이트를 위해 현 코드 섹션을 식별하고, 새 코드 섹션을 현 코드 섹션과 함께 배열하는 것(단계 1008)은 식별된 현 코드 섹션을 새 코드 섹션으로 대체하는 과정을 포함한다.
- <88> 다수의 제 1 심벌 라이브러리들을 다수의 제 2 코드 섹션에 배열하는 것(단계 1001b)은 코드 섹션의 시작점에서 심벌 라이브러리들을 시작하게 하는 과정을 포함한다. 무선 장치 시스템 소프트웨어를 다수의 현 코드 섹션에 저장하는 것(단계 1002)은 해당 시작 어드레스에 코드 섹션 시작점을 저장하는 과정을 포함한다. 그후, 이 방법

은 추가 단계를 포함한다. 단계 1003a는 코드 섹션 식별자를 해당 시작 어드레스와 상호참조시키는 코드 섹션 어드레스 테이블을 관리한다. 다수의 제 1 심벌 라이브러리들을 다수의 제 2 코드 섹션으로 배열하는 것(단계 1001b)은 심벌들을 코드 섹션 시작 어드레스로부터 각각 이격되게 배열하는 과정을 포함한다. 그후, 단계 1003b는 심벌 식별자들을 해당 오프셋 어드레스와 해당 코드 섹션 식별자와 상호참조시키는 심벌 오프셋 어드레스 테이블을 관리한다.

- <89> 해당 시작 어드레스에 코드 섹션 시작점을 저장하는 것은 서브단계들을 포함한다. 단계 1002a는 연속적으로 어드레싱되는 다수의 제 2 메모리 블록을 생성한다. 단계 1002b는 각각의 메모리 블록을 해당 코드 섹션으로 식별한다. 단계 1002c는 식별된 메모리 블록에 코드 섹션들을 저장한다.
- <90> 다수의 제 1 심벌 라이브러리들을 다수의 제 2 코드 섹션들로 배열하는 것(단계 1002)은 배열된 심벌 라이브러리들을 수용하도록 코드 섹션의 크기를 조정하는 과정을 포함한다. 연속적으로 어드레싱되는 다수의 제 2 메모리 블록을 생성하는 것(단계 1002a)은 해당 코드 섹션을 수용하도록 메모리 블록의 크기를 조정하는 과정을 포함한다. 통상적으로, 해당 코드 섹션을 수용하도록 메모리 블록의 크기를 조정하는 것은 배열된 심벌 라이브러리보다 큰 크기를 수용하도록 코드 섹션의 크기를 조정하는 과정을 포함한다. 현 코드 섹션의 크기를 조정하는 것(단계 1006)은 크기조정된 해당 코드 섹션들이 저장되는 메모리 블록의 크기를 조정하는 과정을 포함한다.
- <91> 예를 들어, 단계 1004에서 새 코드 섹션을 수신하는 것은 제 1 크기의 새 코드 섹션을 수신하는 과정을 포함한다. 단계 1005a에서 업데이트를 위해 현 코드 섹션을 식별하는 것은 제 1 크기보다 작은 제 2 크기의 현 코드 섹션을 식별하는 과정을 포함한다. 그후, 단계 1006에서 크기조정된 해당 코드 섹션들이 저장되는 메모리 블록의 크기를 조정하는 것은 서브단계들을 포함한다. 단계 1006a는 식별된 현 코드 섹션에 관련된 메모리 블록의 크기를 제 1 크기로 증가시키고, 단계 1006b는 해당 메모리 블록에 저장된 식별된 현 코드 섹션을 새 코드 섹션으로 대체한다.
- <92> 크기조정된 코드 섹션이 저장되는 메모리 블록의 크기를 조정하는 것(단계 1006)은 해당 코드 섹션 내에 배열되는 심벌 라이브러리들의 크기를 측정함(단계 1005c)에 따라, 코드 섹션 내에 배열되는 심벌 라이브러리 크기를 보다 가깝게 일치시키도록 코드 섹션의 크기를 조정하는 과정을 포함한다. 코드 섹션 내에 배열되는 심벌 라이브러리 크기를 보다 가깝게 일치시키도록 코드 섹션의 크기를 조정하는 것(단계 1006)은 차후 코드 섹션 크기 조정 및 업데이트 동작에 대하여 최적으로 코드 섹션의 크기를 조정하는 과정을 포함한다.
- <93> 단계 1005b는 코드 섹션 어드레스 테이블로부터 시작 어드레스를 이용하여, 현 코드 섹션 크기를 측정한다. 단계 1005c는 심벌 오프셋 어드레스 테이블로부터 심벌 오프셋 어드레스를 이용하여, 해당 코드 섹션 내에 배열된 심벌 라이브러리들의 크기를 측정한다. 즉, 코드 섹션 크기와 심벌 라이브러리들의 크기가 측정된다. 이 정보는 각각의 코드 섹션 내 비사용 영역의 양을 연산할 수 있게 한다.
- <94> 일부 태양에 따르면, 단계 1001a에서 시스템 소프트웨어를 다수의 제 1 심벌 라이브러리로 형성하는 것은 심벌 라이브러리들의 종료를 표시하기 위해 종료 심벌을 형성하는 과정을 포함한다. 그후, 해당 코드 섹션 내에 배열되는 심벌 라이브러리들의 크기를 측정하는 것(단계 1005c)은 심벌 라이브러리들의 크기를 측정하기 위해 종료 심벌들의 오프셋 어드레스를 이용하는 과정을 포함한다. 대안으로, 시스템 소프트웨어를 다수의 제 1 심벌 라이브러리로 형성하는 것(단계 1001a)은 심벌 라이브러리들의 크기를 표시하도록 크기 심벌을 형성하는 과정을 포함한다. 그후, 해당 코드 섹션 내에 배열되는 심벌 라이브러리들의 크기를 측정하는 것(단계 1005c)은 심벌 라이브러리들의 크기 측정을 위해 크기 심벌에 액세스하는 과정을 포함한다.
- <95> 단계 1005d는 새 코드 섹션의 크기를 측정한다. 단계 1005e는 해당 코드 섹션 내에 배열되는 심벌 라이브러리들의 크기 측정에 따라(단계 1005c), 그리고 새 코드 섹션의 크기 측정에 따라(단계 1005d), 새 코드 섹션이 현 코드 섹션과 함께 배열될 수 있는 지를 결정한다.
- <96> 발명의 일부 태양에 따르면, 새 코드 섹션이 현 코드 섹션과 함께 배열될 수 있는 지를 결정하는 것(단계 1005e)은 비사용 메모리 블록의 크기를 결정하는 과정을 포함한다. 그후, 새 코드 섹션을 현 코드 섹션과 함께 배열하여 무선 통신 장치용 업데이트된 시스템 소프트웨어를 형성하는 것(단계 1008)은, 비사용 메모리 블록의 크기가 새 코드 섹션 크기 이상일 경우, 비사용 메모리 블록에 새 코드 섹션을 저장하는 과정을 포함한다.
- <97> 단계 1005f는 코드 섹션 크기를 연산한다. 즉, 새 코드 섹션을 현 코드 섹션과 함께 배열할 수 있게 하는 새 코드 섹션 크기에 대한 결정이 이루어진다. 단계 1007a는 코드 섹션 크기 연산에 따라, 치밀화 스케줄을 발생시킨다. 단계 1007b는 코드 섹션을 파일 시스템 섹션으로 일시적으로 이동시킨다. 그후, 새 코드 섹션을 현 코드 섹션과 함께 배열하여 무선 장치용 업데이트된 시스템 소프트웨어를 형성하는 것(단계 1008)은 치밀화 스케줄에

따라, 파일 시스템 섹션으로부터 메모리 블록에게로 코드 섹션을 저장하여 연속적 어드레싱을 관리하도록 하는 과정을 포함한다.

- <98> 단계 1006의 코드 섹션 크기 조정 단계에 이어, 단계 1009a는 코드 섹션 시작 어드레스를 변경시킨다. 단계 1009b는, 메모리 블록 내에 저장된 코드 섹션들의 시작 어드레스를 변경시킴에 따라, 업데이트된 코드 섹션 어드레스 테이블을 발생시킨다. 그후, 업데이트된 시스템 소프트웨어를 실행하는 것(단계 1010)은 새 코드 섹션을 현 코드 섹션과 함께 배열한 후 업데이트된 코드 섹션 어드레스 테이블을 이용하는 과정을 포함한다.
- <99> 발명의 일부 태양에 따르면, 단계 1004에서 새 코드 섹션을 수신하는 것은 업데이트된 심벌 오프셋 어드레스 테이블로 새 코드 섹션을 수신하는 과정을 포함한다. 그후, 업데이트된 시스템 소프트웨어를 실행하는 것(단계 1010)은 새 코드 섹션을 현 코드 섹션과 함께 배열한 후 업데이트된 심벌 오프셋 어드레스 테이블을 이용하는 과정을 포함한다.
- <100> 단계 1003c는 코드 섹션 어드레스 테이블과 심벌 오프셋 어드레스 테이블을 휘발성 메모리로 로딩시킨다. 단계 1003d(도시되지 않음)는 상기 로딩후, 시스템 소프트웨어를 실행한다. 단계 1009c는 무선 통신 장치를 리셋시킨다. 단계1009d는 상기 리셋 이후, 업데이트된 코드 섹션 어드레스 테이블과 업데이트된 심벌 오프셋 어드레스 테이블을 휘발성 메모리로 로딩한다. 단계 1010은, 두가지 테이블을 메모리로 로딩시킨 다음, 업데이트된 시스템 소프트웨어를 실행한다.
- <101> 도 11은 도 10a-10b의 방법의 대안의 태양을 설명하는 순서도이다. 도 11은 도 10a 및 10b와 공통인 여러 단계들을 공유한다. 단계 1104에서 새 코드 섹션을 수신하는 것은 코드 섹션 크기조정 명령 및 치밀화스케줄을 포함하는 치밀화 명령 세트를 수신하는 과정을 포함한다. 현 코드 섹션의 크기를 조정하는 것(단계 1106)은 코드 섹션 크기조정 명령에 따라 크기를 조정하는 과정을 포함한다.
- <102> 단계 1107은 코드 섹션들을 파일 시스템 섹션으로 일시적으로 이동시킨다. 그후, 새 코드 섹션을 현 코드 섹션과 함께 배열하여 무선 장치용 업데이트된 시스템 소프트웨어를 형성하는 것(단계 1008)은 치밀화 스케줄에 따라, 파일 시스템 섹션으로부터 메모리 블록에게로 코드 섹션을 저장하여 연속적 어드레싱을 관리하게 하는 과정을 포함한다.
- <103> 단계 1104에서 새 코드 섹션을 수신하는 것은 업데이트된 코드 섹션 어드레스 테이블과 업데이트된 심벌 오프셋 어드레스 테이블로 새 코드 섹션을 수신하는 과정을 포함한다. 그후, 업데이트된 시스템 소프트웨어를 실행하는 것(단계 1110)은 새 코드 섹션을 현 코드 섹션과 함께 배열한 후 업데이트된 코드 섹션 어드레스 테이블과 업데이트된 심벌 오프셋 어드레스 테이블을 이용하는 과정을 포함한다.
- <104> 소프트웨어 업데이트 과정을 돕도록 무선 통신 장치의 시스템 소프트웨어 구조를 재조직하기 위한 시스템 및 방법이 위와같이 제시되었다. 이 시스템은 코드 섹션에서의 심벌 라이브러리 배열 때문에, 또한, 심벌 라이브러리 내 심벌들의 오프셋 어드레스와 메모리 내 코드 섹션들의 시작 어드레스에 액세스하기 위한 테이블의 도움으로 쉽게 업데이트가능하다. 이 라이브러리 배열 및 상호참조 테이블의 몇가지 예가 디스플레이 기능을 위해 제시되었으나, 본 발명은 이러한 몇가지 예에 제한되지 않는다.

**도면의 간단한 설명**

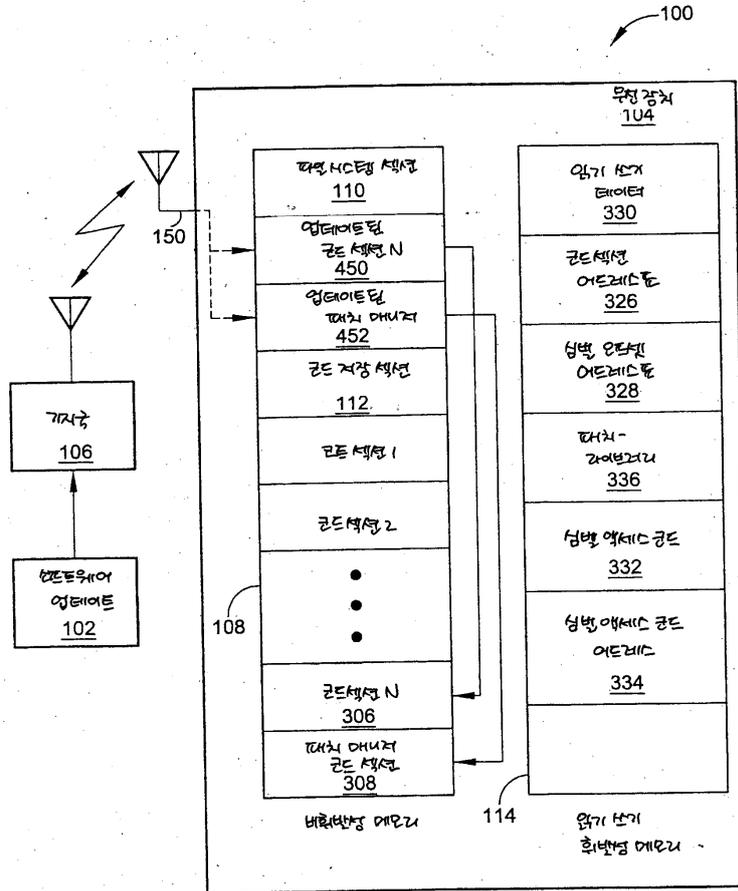
- <13> 도 1은 전체 무선 장치 소프트웨어 관리 시스템의 블록도표.
- <14> 도 2는 에어 링크 인터페이스를 통해 명령 세트의 설치를 강조한, 소프트웨어 관리 시스템의 블록도표.
- <15> 도 3은 무선 통신 장치에서 소프트웨어 명령을 재조직하기 위한 본 발명에 따른 시스템을 도시하는 블록도표.
- <16> 도 4는 무선 장치 메모리의 블록도표.
- <17> 도 5는 도 3의 코드 섹션 어드레스 테이블을 보여주는 표.
- <18> 도 6은 심벌들로 구성된 도 3의 심벌 라이브러리 1의 상세한 도면.
- <19> 도 7은 도 3의 심벌 오프셋 어드레스 테이블을 보여주는 표.
- <20> 도 8은 일례의 치밀화 동작 결과를 보여주는 블록도표.
- <21> 도 9a-c는 일례의 치밀화 스케줄의 동작을 설명하는 블록도표.
- <22> 도 10a-b는 무선 통신 장치 메모리에 저장되는 소프트웨어 명령들을 재조직하기 위한 본 발명에 따른 방법을 설

명하는 순서도.

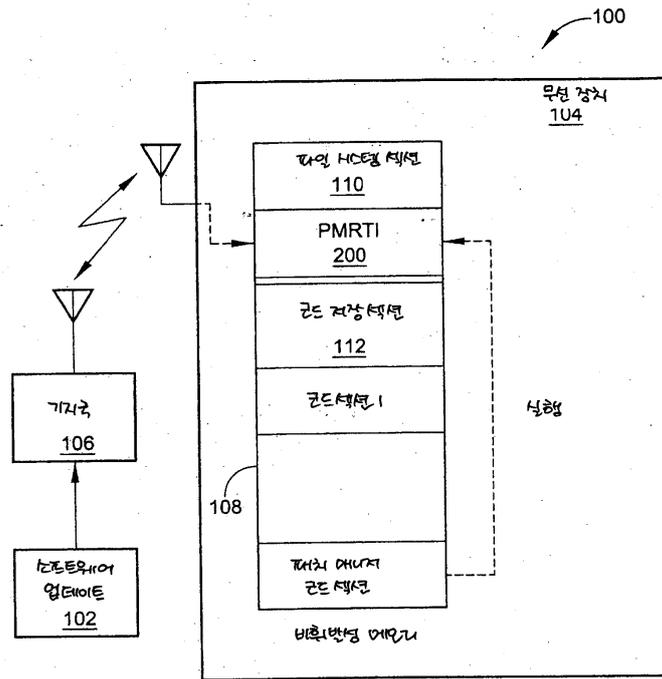
<23> 도 11은 도 10a-b의 방법의 대안의 태양을 설명하는 순서도.

도면

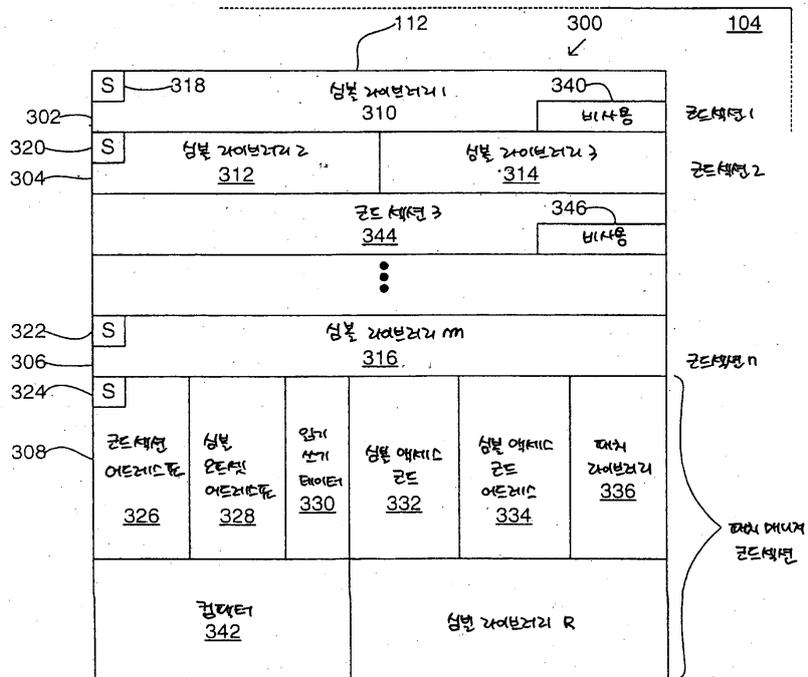
도면1



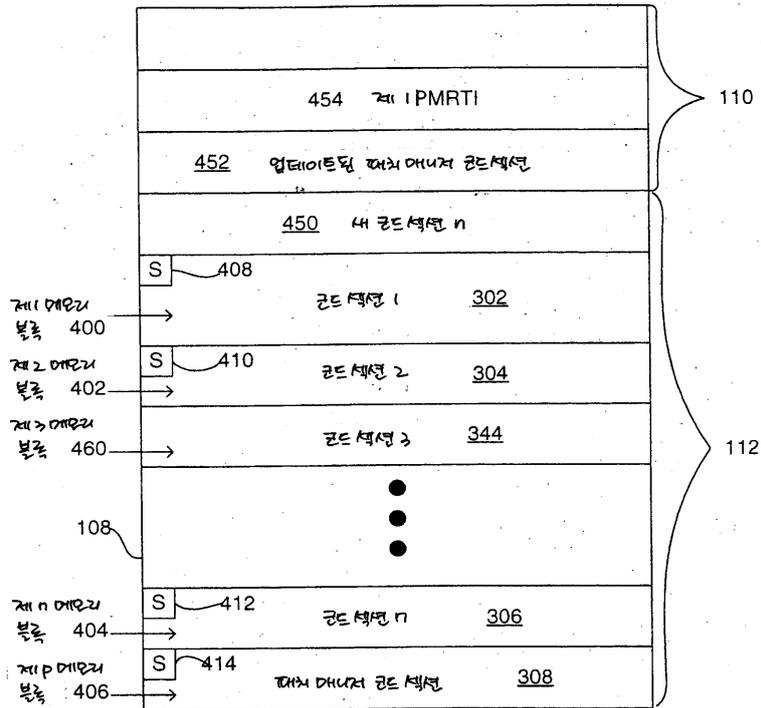
도면2



도면3



도면4

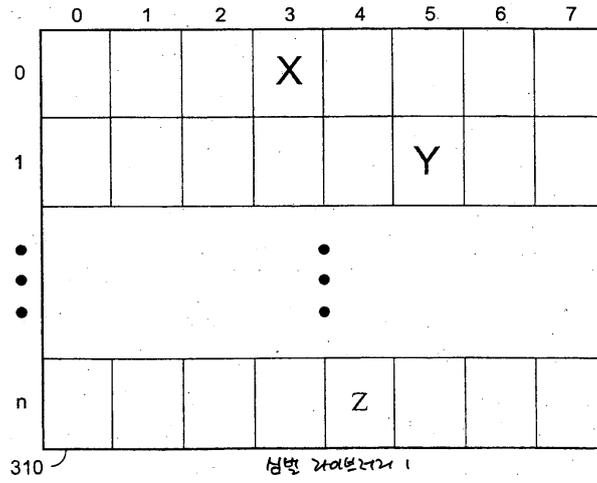


도면5

326

코드 섹션 어드레스표	
변명자	어드레스
CS_1	시작 어드레스 1 (00100)
CS_2	시작 어드레스 2 (00200)
	• • •
CS_N	시작 어드레스 N (00700)
PM	시작 어드레스 P (01000)

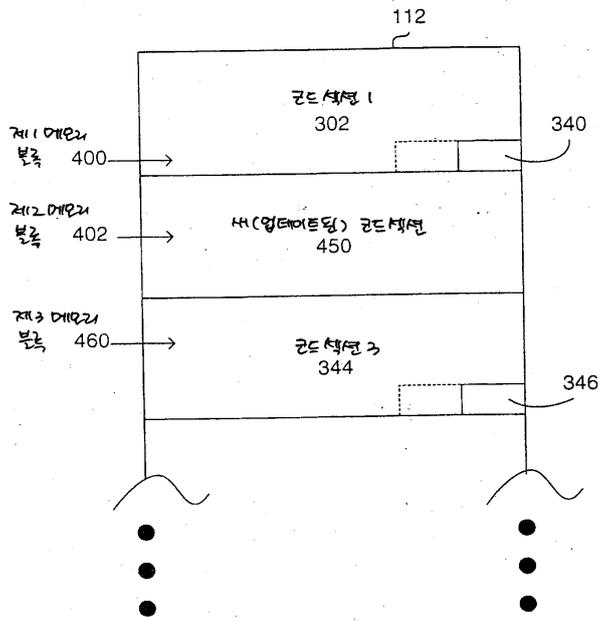
도면6



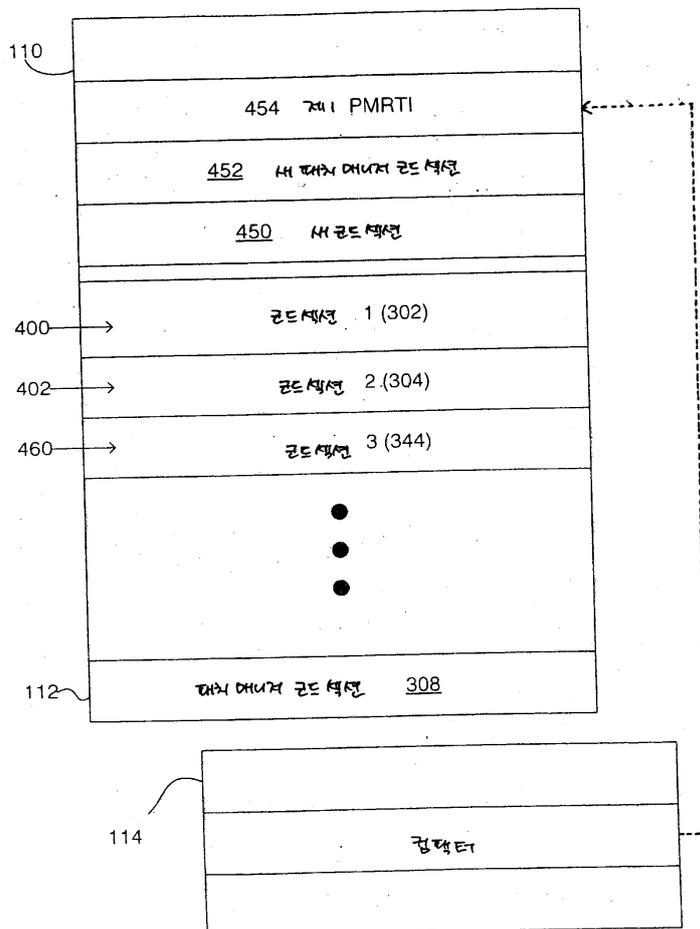
도면7

심벌 인덱스 어드레스들		
심벌 ID	코드 세션 ID	인덱스
X_1	CS_1	03
Y_1	CS_1	15
P_1	CS_2	11
Q_1	CS_2	33
AA_3	CS_2	47
•		
•		
•		

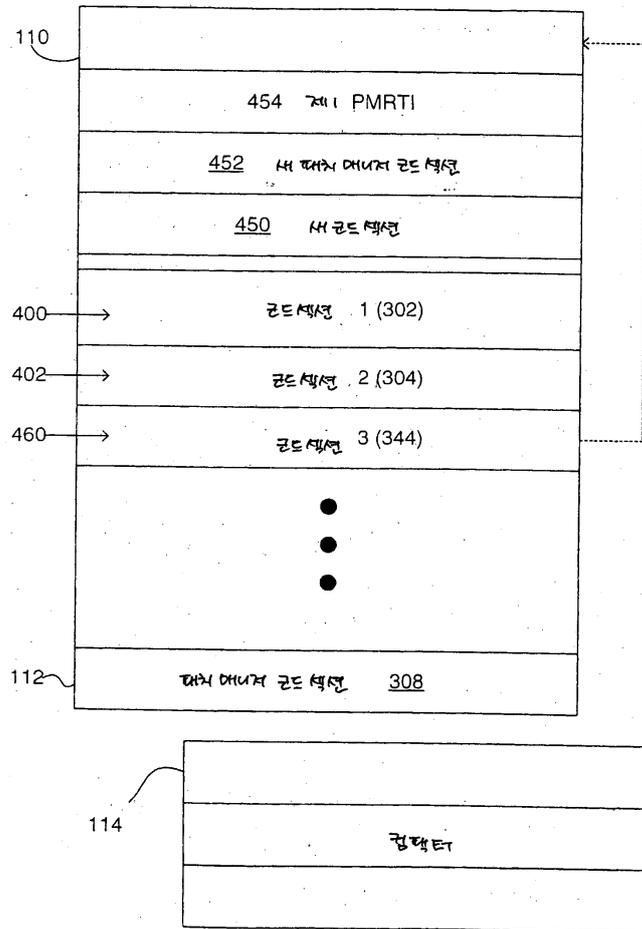
도면8



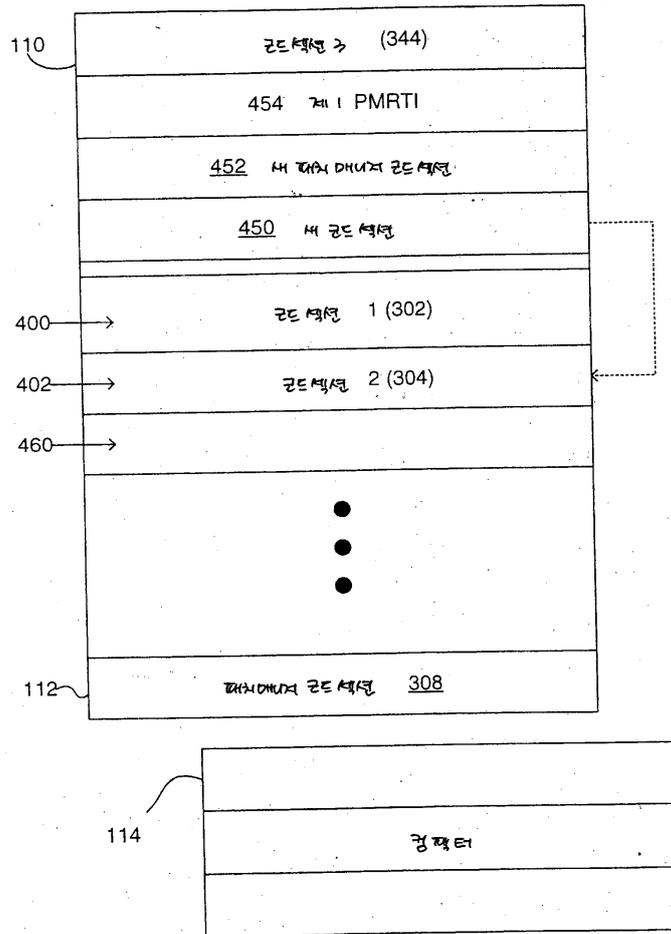
도면9a



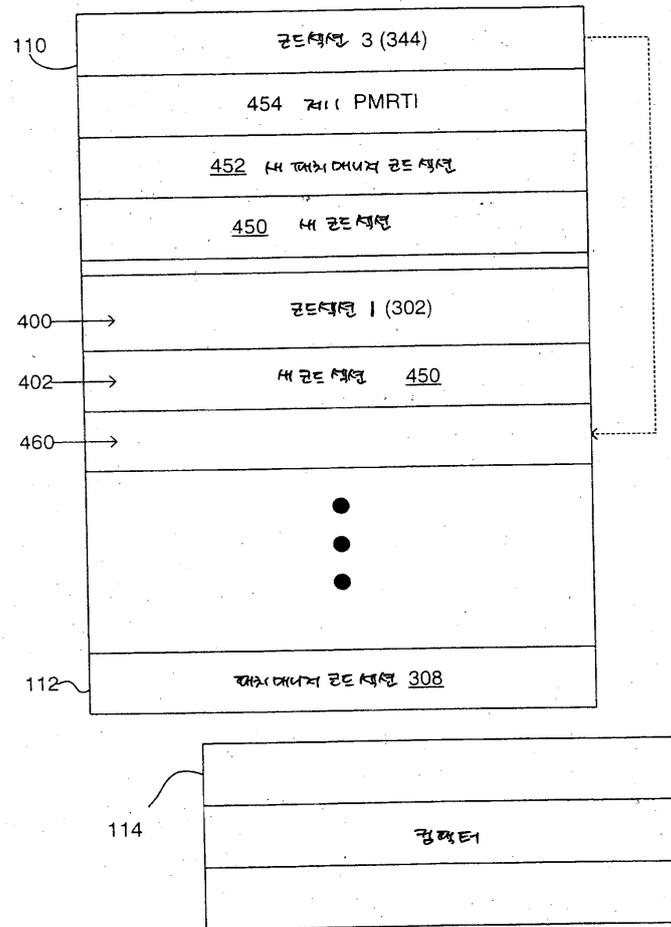
도면9b



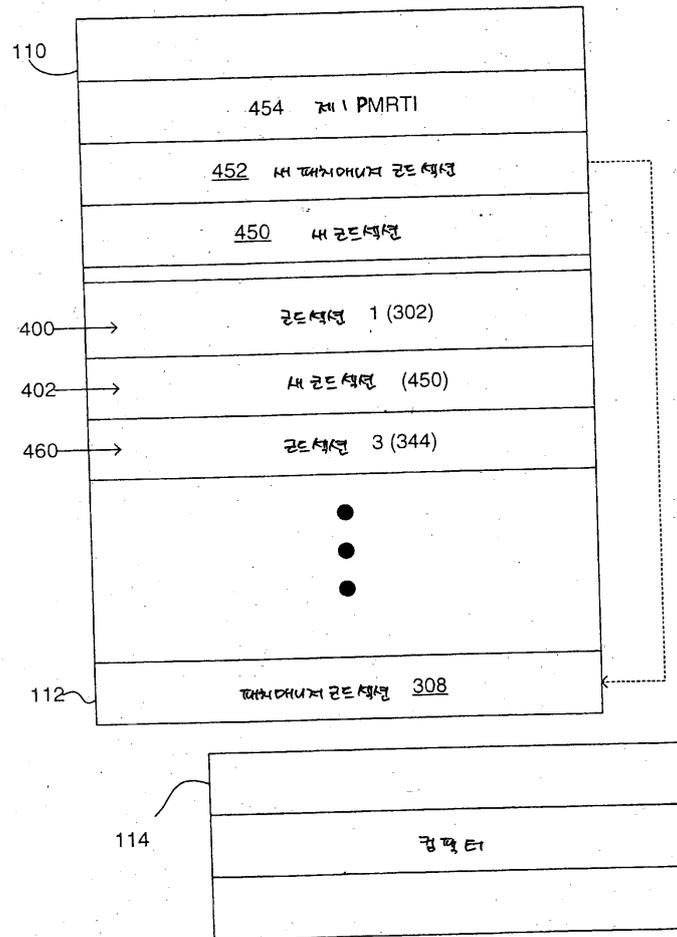
도면9c



도면9d



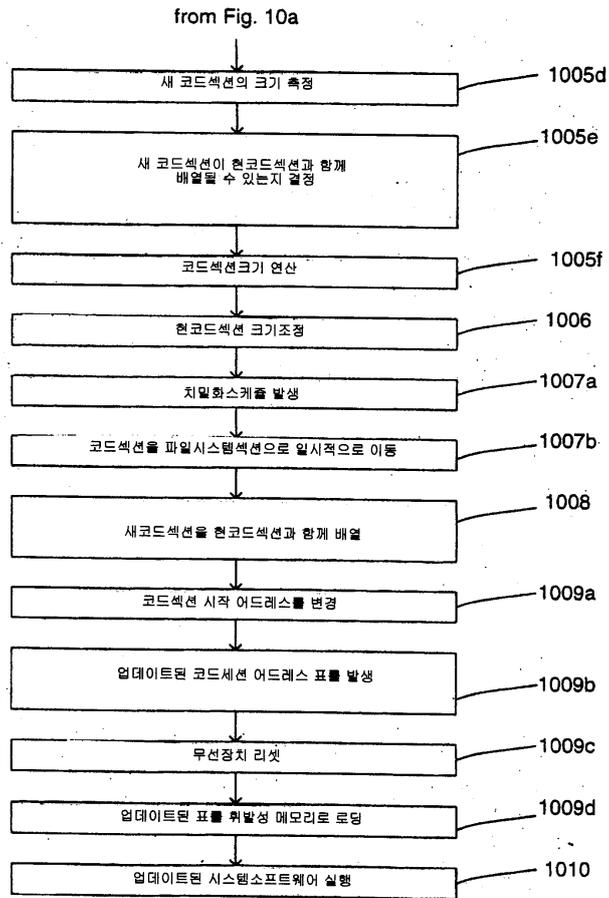
도면9e



도면10a



도면10b



도면11

