(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(51) Int. Cl.[6]
G06K 19/06

(11) 공개번호 특2001-0006507
(43) 공개일자 2001년01월26일

(21) 출원번호 10-1999-7009597
(22) 출원일자 1999년10월16일
번역문제출일자 1999년10월16일
(86) 국제출원번호 PCT/US 98/07905
(87) 국제공개번호 WO 98/47101
(86) 국제출원출원일자 1998년04월16일
(87) 국제공개일자 1998년10월22일
(81) 지정국 EP 유럽특허 : 오스트리아 벨기에 스위스 독일 덴마크 스페인 프랑스 영국 그리스 아일랜드 이탈리아 룩셈부르크 모나코 네덜란드 포르투칼 스웨덴 핀랜드 사이프러스
국내특허 : 중국 일본 대한민국
(30) 우선권주장 08/842,644 1997년04월16일 미국(US)
08/914,324 1997년08월19일 미국(US)
09/021,608 1998년02월10일 미국(US)
(71) 출원인 인터멕 테크놀로지즈 코포레이션 토마스 오. 밀러
미합중국 워싱턴주 에베레트 제36 애브뉴 웨스트 6001
(72) 발명자 액리에이치.스프라그
미국워싱톤98103시애틀화이트먼애비뉴노스4253
위클로프크리스토퍼에이.
미국워싱톤98208에베레트사우스이스트99스트리트3531
(74) 대리인 이병호

심사청구 : 없음

(54) 기계-판독 가능한 심볼로지 및 그의 인쇄 및 판독 방법 및 장치

요약

전형적인 실시예에서 본 발명의 신규 바 코드 심볼로지는 코드 93과 유사하게 9개의 모듈러스 내에 3개의 바들(및 스페이스들)을 사용한다. 여러 개의 특수 모드 문자들을 포함하는 53개 데이터 문자들이 정의된다. 특정 루틴과 함께 이들 특수 모드 문자들을 사용함으로써, 3개의 기호 문자들은 2개의 8-비트 바이트, 또는 하나의 16비트 워드를 나타낼 수 있다. 결과적으로, 심볼로지는 컴퓨터 프로세싱에 사용하기 위해 8-비트 바이트를 효율적으로 인코딩할 수 있거나, 또는 유니코드 등의 16-비트 문자 세트를 인코딩할 수 있다. 심볼로지는 확장된 채널 해석(ECI) 번호를 인코딩하고, 다중 숫자 압축 모드를 제공하고, 기타 특징들 뿐만 아니라 단일 모드 문자를 사용하는 구축된 부록을 제공한다. 또한, 이 심볼로지는 기호 내에 에러 정정의 사용을 나타내는 특수 형상의 프래그 문자에 의한 에러 정정을 포함한다.

대표도

도2

색인어

바 코드 심볼로지, 기호 문자, 확장된 채널 해석(ECI) 번호

명세서

기술분야

본 발명은 신규한 기계-판독 가능한 심볼로지 및 그 심볼로지에 따라 기호들을 판독 또는 인쇄하기 위한 장치 및 방법에 관한 것이다.

배경기술

바 코드 심볼로지는 최초로 커모드(Kermode)에 의해 미국 특허 제1,985,035호에 개시되었고, 이후 1930년대에 Westinghouse에 양도된 Young의 미국 특허 제2,020,925호에서 간단히 확대되었다. 이들 초기 심볼로지는 보다 큰 반사율의 소자, 또는 "스페이스"에 의해 분리되는 낮은 반사율의 가지 각색의 단일폭 소자, 또는 "바"를 발생시킴으로써 인쇄되었다. "소자"란 바 또는 스페이스이

다. 현재 사용되는 이들 초기 심볼로지 및 많은 "바 코드 심볼로지"는 주어진 기호의 데이터가 하나의 축 또는 방향을 따라 디코딩되기 때문에 "선형 심볼로지"라 칭할 수 있다. 선형 심볼로지 등의 심볼로지는 특정 데이터 문자를 인코딩하도록 유일한 그룹의 패턴을 형성하는 바 및 스페이스를 선택하여 병렬 배열시키는 "기호 문자"로서 "데이터 문자"(예, 인간이 판독 가능한 문자)를 인코딩한다. "데이터 문자"는 인간이 판독 가능한 문자를 포함할 뿐만 아니라, 특정 기능의 데이터를 제공하는 시작, 종료 또는 시프트 문자 등의 특정 기능 문자를 포함한다. 소정의 폭 내의 바 및 스페이스의 각각의 유일한 그룹 또는 패턴은 특정 기호 문자, 및 그에 따른 특정 데이터 문자 또는 문자들을 정의한다.

공지된 U.P.C. 심볼로지는 일반적으로 (7,2) "n,k 코드"로서 기재될 수 있다. "n,k 코드"는 각각의 기호 문자가 "k" 번호의 바 및 스페이스를 갖고, 그의 전체 길이가 "n" 모듈 길이인 심볼로지로서 정의된다. 따라서, U.P.C. 심볼로지는 각각의 기호 문자에 2개의 바 및 2개의 스페이스를 인코딩하고, 각각의 기호 문자는 7 모듈 길이이다. "모듈"은 바 코드 심볼로지에서 가장 좁은 명목상의 폭 유니트의 측정치이다. "명목상"은 인쇄 에러 등과 무관하게, 특정 파라메터의 의도된 값을 언급한다. 통상적인 계수 기술에 따라, 가능한 기호 문자들의 수는 7개의 모듈에서, 트랜지션이 발생할 수 있는 경우 6개의 위치가 존재하고, 2개의 바 및 2개의 스페이스에 대해, 3가지 내부 트랜지션이 존재하는 것을 이해함으로써 발견될 수 있다. 따라서, U.P.C. 심볼로지에 대한 유일한 기호 문자들의 수는 20과 동일하게 단순히 6 선택 3이다. (11,3) 심볼로지인 코드 128 심볼로지에 따라, 252개의 독특한 기호 문자들이 이용된다(10 선택 5).

U.P.C., EAN, 코드 11 및 코다바로서 공지된 바 코드 심볼로지는 단지 숫자 데이터 문자, 및 "+"와 "-" 등의 몇몇 특수 문자를 지원하는 모든 바 코드 심볼로지 표준이다. U.P.C. 심볼로지는 표준 용도에서 공업적으로 채택된 것중에서(소비자 상품), 공업 표준 뿐만 아니라 바 코드 표준이다. 바 코드 표준 코드 39는 개발된 첫 번째 영숫자 바 코드 심볼로지이다. 그러나, 그것은 43개의 문자로 제한된다.

코드 93은 코드 39에 비해 개선된 것이다. 코드 93은 4개의 소자 폭을 사용하는 연속적인 바 코드 심볼로지이다. 각각의 코드 93 기호는 흑색 또는 백색(바 또는 스페이스)일 수 있는 9개의 모듈러스를 갖는다. 코드 93 표준에서 각각의 기호는 3개의 바 및 3개의 스페이스(6개의 소자)를 포함하고, 그의 전체 길이는 9개의 모듈러스 길이이다. 기호당 9개의 모듈러스 및 3개의 바를 갖는 코드 93은 56개의 가능한 문자(8 선택 5)를 갖는 (9,3) 심볼로지이다. 에지-에지 디코딩 이유 때문에, 코드 93 심볼로지 표준은 단지 48개의 독특한 기호를 정의하고, 따라서, 그의 문자 세트 + 시작/종료 코드로 47개 문자를 정의할 수 있다. 47개 문자는 숫자 문자 0-9, 영문자 A-Z, 일부 부가적 기호 및 4개의 시프트 코드를 포함한다.

컴퓨터 업계는 자체의 문자 인코딩 표준, 즉, 정보 교환을 위한 미국 표준 코드(ASCII)를 사용한다. ASCII는 128개의 문자 및 기호를 포함하는 문자 세트를 정의한다. ASCII에서 각각의 문자는 독특한 7-비트 코드로 나타낸다. 코드 39 및 코드 93은 50개의 문자 보다 더 적은 수로 제한되고, 이들 표준은 각각의 ASCII 문자를 유일하게 나타내기에 부적절하다. 그러나, 코드 93에서 4개의 시프트 코드는 이러한 표준이 128개 ASCII 문자 모두를 모호하지 않게 나타낼 수 있게 한다. 한가지 결점은 일련의 2가지 코드 93 기호들이 단일 ASCII 문자를 나타내기 위해 필요하다는 것이다. 따라서, ASCII 문자 세트에서 문자들을 나타내는 바 코드 라벨은 코드 93 문자 세트에서 문자들을 나타내는 라벨의 길이의 2배이다.

코드 128 등의 신규 바 코드 심볼로지 표준은 완전한 ASCII 문자 세트를 인코딩하기 위해 개발되었으나, 이들 표준은 특정 문자를 나타내기 위해 시프트 코드 또는 기타 선행 기호들을 요하는 것을 포함하는 특정 단점을 경험하게 된다. 모든 이들 심볼로지는 전체 ASCII 문자 세트를 처리하기 위해 증가된 처리 시간 및 경비를 필요로 한다.

컴퓨터 업계는 ASCII 문자 세트의 한계를 넘어서 성장하였다. 컴퓨터 시장이 성장함에 따라, ASCII 문자 세트로 정의하지 못하는 부가적인 언어를 지원할 필요성 역시 증가하였다. 신규 문자 세트는 관련된 언어로 문자들의 클러스터를 수용하도록 개발되었다. 원시 7-비트 ASCII 문자 세트는 8비트로 확대됨으로써, 부가적인 128개의 문자 또는 데이터 값을 제공한다. 부가적인 문자에 대해 허용되는 데이터 값의 이와 같이 부가적인 128개 세트("상위 128" 또는 "확장된 ASCII")는 나타낼 관련 로맨스 언어(즉, 불어, 독일어, 스페인어 등)에 존재한다. 8-비트 데이터를 인코딩할 수 있는 유일한 선형 심볼로지는 "강한 시작/종료 문자 및 에러 검출 도식을 이용하는 완전한 문자 세트를 갖는 단일 폭 바 코드 심볼로지(Single Width Bar Code Symbology With Full Character Set Utilizing Robust Start/Stop Characters and Error Detection Scheme)"라는 표제의 발명자들의 미국 특허 제5,619,207호에 기재된 코드 128 및 "코드 53"이다. 코드 28 및 코드 53 모두는 단일 또는 이중 기능 시프트 문자를 사용함으로써 8-비트 데이터를 인코딩하고, 따라서 모든 바이트 값은 데이터 문자가 인코딩되기 전에 분석되어야 하기 때문에, 증가된 처리 시간 및 경비를 요하게 된다.

그러나, 컴퓨터 시장이 국제적으로 성장함에 따라, 훨씬 더 많은 언어들이 문자 세트에 포함될 필요가 있게 되었다. 특히, 아시아의 시장은 컴퓨터 상에서 사용될 수 있는 수천 개의 독특한 문자들을 지원하는 문자 세트를 요구하고 있다. 이들 문자들 각각을 유일하게 정의하기 위해, 16-비트 인코딩 표준이 필요하다.

유니코드(Unicode), JISC-6226-1983 등의 여러 가지 16-비트 인코딩 표준, 및 기타 표준이 현재 개발되고 있다. 유니코드 문자 인코딩 표준은 고정 길이의 균일한 텍스트 및 문자 인코딩 표준이다. 유니코드 표준은 65,536개에 달하는 문자를 포함할 수 있고, 현재 그리스어, 히브루어, 라틴어, 일본어, 중국어, 한국어, 및 대만어를 포함하는 세계의 스크립트 상으로 매핑되는 28,000개 이상의 문자를 포함한다. 유니코드 표준은 ASCII 문자 세트로 모델링되고 있다. 유니코드 문자 코드는 언어와 무관하게 일정하게 16비트 길이이므로, 임의의 언어에서 임의의 문자를 특정하기 위해 어떠한 에스케이프 시퀀스 또는 제어 코드도 필요치 않다. 유니코드 문자 인코딩은 문자들,

영문자들 및 표의 문자들을 동일하게 다룸으로써, 이들이 여러 가지 컴퓨터 용도에 동시적으로 및 동일하게 용이하게 사용될 수 있다. 텍스트를 디스플레이할 수 없거나 또는 인쇄할 수 없는 문자를 나타내기 위해 유니코드 문자 인코딩을 사용하는 컴퓨터 프로그램은 신규 스크립트 또는 문자들이 도입될 때 변경되지 않은 채로 유지될 수 있다.

신규 컴퓨터 오퍼레이팅 시스템은 이와 같이 포괄적인 16-비트 코드 표준, 예를 들면 워싱턴주 레드먼드 소재 마이크로소프트사가 제조한 WINDOWS NT를 지원하기 시작한다. 그러나, 데이터 수집 업계는 컴퓨터 업계와 보조를 맞추는 데 실패하였다. 현재 16-비트 컴퓨터 문자 코드를 바 코드 기호로 용이하게 인코딩하기 위한 어떠한 시스템도 존재하지 않는다. 따라서, 데이터 수집 업계, 특히 바 코드 심볼로지에 있어서 이들 16-비트 컴퓨터 문자 표준을 지원할 필요가 있다.

더욱이, 대부분의 영숫자 바코드 심볼로지는 긴 시리즈의 숫자 또는 하위 케이스의 영문자를 인코딩하기 위해 사용될 때 비능률적이다. 예를 들면, 코드 93을 사용하는 일련의 십진수를 인코딩할 때, 26개의 영문자를 반영하는 26개의 바코드 기호가 사용되지 않는다. 따라서, 이들 영숫자 바코드 심볼로지가 긴 시리즈의 숫자를 보다 효율적으로 나타내게 할 필요가 있다. 또한, 많은 심볼로지는 에러 정정 특징이 결여되었다.

한기지 공지된 심볼로지인 IPC 심볼로지는 또 다른 기호가 인접한지 여부 및 어떠한 방식으로든지 IPC 기호에 관련된 데이터를 포함하는지 여부를 지시한다. IPC 심볼로지에 대한 데이터 세트에서 패리티 배열은 이와 같이 인접한 기호의 존재를 지시한다. IPC 심볼로지는 U.P.C. 심볼로지 등의 Nniform Code Council(UCC) 용도에 사용된 고정된 폭의 숫자 심볼로지이다. IPC 심볼로지는 고정된 폭이기 때문에 고정된 수의 문자들 만이 하나의 기호 내에서 인코딩될 수 있다. 더욱이, 숫자 데이터 문자들 만이 이러한 기호들에서 인코딩될 수 있다. 결과적으로, IPC 심볼로지는 최소 데이터 수집 용도로만 제한된다.

발명의 상세한 설명

본 발명은 상기 문제점들을 해결하고, 부가적인 이익을 제공한다. 본 발명의 일 실시예에서, 그룹 시프트 특징은 하위 케이스의 영문자들, 숫자 문자들 및 기타 하위 ASCII 문자들의 스트링을 예를 들면 이들 하위 케이스의 영문자들/숫자 문자들 10개의 하나의 그룹에 대해 기호 문자당 10.8 모듈 내에서 효율적으로 디코딩될 수 있게 한다.

다른 실시예에 따라, 에러 정정 또는 특수 형상의 프래그는 기호 내의 에러 정정 문자들을 지시하기 위해 사용될 수 있다. 특수 형상의 프래그 문자는 메시지 길이 지시기 문자 이후에 기호의 중앙에 위치된다. 전형적인 실시예에서, 에러 정정을 사용하는 기호는 2개의 에러 또는 3개의 소거가 기호 내에서 정정되게 하는 5개의 에러 정정 문자를 포함한다. 더욱이, 기호의 시작 또는 끝 부분들이 정정될 수 있고, 기호는 절반으로 인코딩될 수 있다.

다른 실시예에서, 심볼로지는 임의의 바이트 값에 대해 균일하게 작용하는 단순한 바이트 코드화 모드를 기재함으로써 8-비트 및 16-비트 데이터를 인코딩하는 복잡한 방법을 피한다. 전형적인 실시예에서, 본 심볼로지는 코드 93 심볼로지과 유사하므로, 기호 문자들은 단지 9개의 모듈러스 길이이다. 3개의 기호 문자들은 2개의 8-비트 바이트를 인코딩한다. 따라서, 하나의 바이트는 바이트 값과 무관하게 약 13.5 모듈을 필요로 한다. 데이터 문자들을 인코딩하기 위한 모듈러스의 수의 추가의 감소는 디지트 또는 베이스 코드 93 데이터 문자들의 스트링 등의 전형적인 심볼로지에 따라 허용된다.

본 발명의 실시예에서, 확장된 채널 해석(ECI) 번호는 공식적으로 인코딩된다. 결과적으로, 판독기가 결합될 수 있는 호스트 컴퓨터 시스템은 이 호스트 컴퓨터에 의해 사용된 기본적인 문자 세트 또는 용도와 무관하게 세계 어느곳에서나 코드된 메시지를 유일하게 디코딩할 수 있다. 16-비트 문자들은 전형적인 심볼로지에 따라 3개의 문자로 나타내며, 다른 모드에서 2개의 8-비트 바이트는 3개의 기호 문자들로 나타낸다. 3개의 기호 문자들 내에 2개의 8-비트 바이트를 나타냄으로써, ISO 시리즈 8859-1-8859-9 등의 비교적 적은 국제 문자 세트 뿐만 아니라 확장된 ASCII 데이터 문자들이 공식적으로 인코딩될 수 있다.

광범위한 의미에서, 본 발명은 복수개의 기호 문자들을 갖는 기계-판독 가능한 심볼로지를 구현한다. 기계-판독 가능한 심볼로지는 적어도 하나의 에러 정정 기호 문자가 기호 문자들의 그룹 내에 존재하는 것을 나타내는 에러 정정 플래그 기호 문자를 포함한다.

또한, 본 발명은 복수개의 기호 문자들을 갖는 기계-판독 가능한 심볼로지를 구현하고, 여기서, 기호 문자들의 그룹은 인쇄 가능한 기호를 형성한다. 기계-판독 가능한 심볼로지는 인쇄 가능한 기호 내의 제1 위치에 제1 기능을 나타내는 특수 형상의 프래그 기호 문자를 포함한다. 특수 형상의 프래그 기호는 인쇄 가능한 기호 내의 제2 위치에 제 2 기능을 지시한다. 더욱이, 심볼로지는 코드 93 시작 및 종료 문자, 및 표준 코드 93 심볼로지에서의 값들 이상의 문자 값들을 갖는 기호를 식별하기 위한 2개의 독특한 시작 및 종료 문자, 에러 정정을 갖는 기호, 및 전형적으로 현재 기호에 인접하거나 또는 그 근처에 위치하는 관련된 또는 짝 기호들을 갖는 기호들의 조합을 사용할 수 있다. 또한, 본 발명은 여러 국면에서 기계-판독 가능한 기호들을 인쇄 및 판독하기 위한 방법 및 장치 및 본 발명의 기타 실시예 및 국면을 구현한다.

도면의 간단한 설명

도 1은 본 발명의 전형적인 실시예에 따라 인쇄 또는 판독되는, 인간이 판독 가능한 문자들과 함께, 바 코드 기호 문자들을 갖는 라벨의 예이고,

도 2는 본 발명의 전형적인 심볼로지에 대한 기호 값들 및 연관된 기호 문자들 및 데이터 문자들을 보여주는 테이블이며,

도 3은 전형적인 기호 문자 포맷을 보여주고,

도 4는 그의 관련된 데이터 문자로 식별된 각각의 기호 문자를 갖는, 도 1의 바 코드 기호를 보여주며,

도 5는 도 2의 심볼로지에 따라 인코딩될 수 있는 완전한 ASCII 데이터 문자를 보여주는 테이블이고,

도 6은 본 발명의 바 코드 기호 인쇄 장치의 블록도이며,

도 7은 전형적인 실시예 하의 바 코드 기호들을 인쇄하기 위한 도 6의 인쇄 장치에 의해 수행되는 기본 단계를 보여주는 전형적인 흐름도이고,

도 8은 본 발명의 바 코드 기호 판독 장치의 블록도이며,

도 9는 전형적인 실시예 하의 바 코드 기호들을 판독하기 위한 도 8의 판독 장치에 의해 수행된 기본 단계들을 보여주는 흐름도이고,

도 10은 에러 정정 문자들을 사용하는 93i 기호의 개략도이며,

도 11은 인접한 짝 기호를 갖는 93i 기호의 개략도이고,

도 12는 93i 기호 및 짝 2차원 기호의 블록도이며,

도 13은 93i 기호 및 짝 RF 태그의 블록도이고,

도 14는 93i 기호 위에 위치된 짝 PDF417 기호와 함께 93i 기호를 보여주며,

도 15는 93i 기호의 우측에 위치된 짝 코드 49 기호와 함께 93i 기호를 보여준다.

실시예

본 명세서에 일반적으로 사용된 바와 같이, 하기 정의가 적용된다: "데이터 문자"는 기능 코드, 시프트 코드 등의 판독 불가능한 데이터 뿐만 아니라 기호들, 숫자 문자들, 영문자들 및 표의 문자들을 포함하는 인간이 판독 가능한 문자를 의미하고; "숫자 스트링"은 전형적으로 십진수 디지트 등의 숫자 문자들의 시퀀스를 의미하고; "문자 코드들"은 코드, 전형적으로 숫자를 의미하고, 이는 문자 코드 세트 내의 데이터 문자 및 대응하는 데이터 문자들, 예를 들면 ASCII를 의미하고, 여기서 "8-비트 코드"는 ASCII 표준에서 데이터 문자에 대응하는 확장된 ASCII 코드를 의미하고, "16-비트 코드" 또는 "16-비트 문자 코드"는 유니코드 등의 16-비트 문자 인코딩 표준에서 데이터 문자의 16진수 또는 십진수 표기법을 의미하고; "바 코드 심볼로지"는 데이터 문자 세트를 독특하게 나타내기 위한 기계-판독 가능한 문자들 또는 기호 문자들 세트를 의미하고; "기호 값"은 바 코드 심볼로지에서 데이터 문자를 나타내는 서수 등의 코드를 의미하고; "기호 문자"는 특정 데이터 문자들을 나타내기 위해 바 코드 심볼로지에 사용된 독특한 기하학적 형상 또는 바 또는 스페이스 패턴을 의미하고; "바 코드 표준"은 데이터 수집 용도에 의해 인식되거나 또는 그에 규칙적으로 사용된 바 코드 심볼로지(예, 코드 128, 코드 93)를 의미하고; "카운트"는 바 코드 심볼로지에서 데이터 문자에 대응하는 기호 문자를 판독할 때 사용된 독특한 세트의 전기 신호들을 의미한다.

예를 들면, 16-비트 문자 인코딩 표준 유니코드에서, 데이터 문자 "A"는 16진수 표기법에서 16-비트 코드 "0041"로 나타내고 십진수 표기법에서 "65"로 나타낸다. 데이터 문자 "A"는 바 코드 심볼로지 코드 93에서 "10"의 기호 값을 갖는다. 코드 93에서 기호 값 10은 단일 모듈 폭 스페이스, 단일 모듈 폭 바, 단일 모듈 폭 스페이스, 단일 모듈 폭 바, 및 3개의 모듈 폭 스페이스가 후속하는 2개의 모듈 폭의 패턴을 갖는 기호 문자에 대응한다. 이러한 기호의 인쇄와 연관된 카운트는 일반적으로 각각의 프린터에 대해 독특하고, 열 프린터에 대해, 바들이 프린터를 통해 라벨의 방향에 대해 수직으로 위치된 경우에 그를 지나쳐 시프트된 감열 페이퍼와 같이 프린터의 가열 소자를 적절히 활성화시키기 위해 바들과 스페이스들 간의 트랜지션 사이의 시간격을 나타낼 수 있다. 대안으로, 카운트는 바들이 프린터를 통해 라벨의 방향과 평행할 때 활성화시키기 위한 도트 또는 써멀-프린트 소자를 지시할 수 있다.

일반적으로 본 명세서에서 "93i"라 칭하는 본 발명의 실시예에 따른 신규 바코드 또는 선형 심볼로지는 임의의 16-비트 문자 코드에서 각각의 16-비트 코드를 유일하게 나타내고, ECI 문자를 인코딩할 뿐만 아니라 본 명세서에 상세히 기재된 부가적인 특징을 제공하기 위해 데이터의 바이트 및 워드를 효율적으로 인코딩한다. 도 1은 93i 심볼로지에 따라 인쇄 또는 판독되는 라벨 101의 예를 보여준다. 도 1에 나타낸 바와 같이, 라벨 101은 인쇄된 안간이 판독 가능한 문자들에 대응할 뿐만 아니라 데이터 문자들을 인코딩하는 일련의 바 코드 기호들을 포함한다.

93i 심볼로지는 코드 93과 유사하다. 결과적으로, 93i 심볼로지는 숫자, 영숫자 및 완전한 128 ASCII 문자들을 인코딩한다. 부가적으로, 93i 심볼로지는 확장된 ASCII 문자들 및 16-비트 문자 코드로 나타낸 것들과 같은 모든 국제적 문자 세트를 인코딩한다. 93i 심볼로지는 연속적인 것으로, 아래 기재하는 바와 같이 3개의 바들 및 3개의 스페이스들과 더불어 기호당 6개의 소자를 갖는 기호 구조를 사용한다. 93i 심볼로지에 따른 문자들은 자체-체킹되지 않고, 기호 길이는 가변적이다. 93i 심볼로지는 2개의 기호 체크 문자 또는 임의의 에러 정정 문자를 사용한다. 93i 심볼로지는 비데이터 비용과 동등하게 37 모듈을 사용한다. 중요하게는, 93i 심볼로지는 다음과 같이 데이터 문자 밀도를 허용한다: 즉, 숫자 디지트당 5.4 모듈, 영숫자 데이터에 대해 기호 문자당 9 모듈, 완전한 ASCII 및 확장된 ASCII에 대해 13.5 모듈(ISO8859 8-비트 단일-바이트 코드된 그래픽 문자 세트 표준에 따라 정의됨), 하위 케이스 영문자들 및 기타 하위 ASCII 문자들의 스트링에 대해 10.5 모듈 및 아시아 문자 또는 16-비트 문자 코드 문자들 당 27 모듈. 부가적으로, 본 발명이 심볼로지는 확장된 채널 해석(ECI) 프로토콜(아래 기재됨)을 지원하고, 현존하는 코드

93 심볼로지과 완전히 호환적이다.

도 2는 93i 심볼로지에서 각각의 데이터 문자에 대한 기호 문자 할당을 보여준다. 도 2에서 "값" 칼럼은 각각의 기호 문자에 대한 기호 값을 나타낸다. 본 명세서에 기재된 바와 같이, 기호 값은 체크 또는 에러 정정 문자를 연산하기 위해 사용될 뿐만 아니라, 여러 가지 데이터 압축 방법에도 사용된다. 도 2에서 "문자"는 각각의 기호 문자에 대한 교대하는 바 및 스페이스 패턴을 열거하고, 여기서 "1"은 하나의 모듈에 대응하고, "2"는 2개의 모듈에 대응한다. 각각의 문자는 바로 시작한다. 도 2에서 "데이터" 칼럼은 각각의 기호 문자에 대응하는 베이스 데이터 문자 또는 그 기호 문자의 기능성을 나타낸다. 도 2에 나타낸 바와 같이, 기호 값 00-46, 및 이들의 대응하는 기호 문자들 및 데이터 문자들은 시작 및 종료 기호 문자들 뿐만 아니라 코드 93 심볼로지에서 대응하는 기호 값들, 기호 문자들 및 데이터 문자들과 일치한다.

부가적으로, 93i 심볼로지는 독특한 시작 및 종료 문자들을 사용하고: 즉, 시작 문자는 바 및 스페이스 패턴 2,1,3,1,1,1로 구성되고, 종료 문자는 동일한 바 및 스페이스 패턴으로 구성되지만 말단에 부가적으로 2개의 넓은 바를 포함하는, 즉, 2,1,3,1,1,1,2로 구성된다. 이러한 선택적인 시작 및 종료 문자들은 코드 93 심볼로지로부터 본 발명의 심볼로지를 차별화시키고, 흐릿한 디코딩을 조장하기 위해 문자의 시작 및 종료 시에 넓은 소자를 갖는 문자들을 제공한다. 보다 중요하게는, 2가지 입수 가능한 시작 및 종료 문자들은 4가지 독특한 유형의 기호들이 유효하게 한다.

코드 93 시작 및 종료 문자들 각각으로 시작 및 종료되는 제1 유효 기호 타입은 47 미만의 문자 값을 갖는 모든 기호 문자들을 포함하고, 모든 시프트 문자는 데이터 문자 세트 A, V,..., Z 내의 문자 값이 후속한다. 제2 유효 기호 타입은 코드 93 시작 문자로 시작하고, 46을 초과하는 적어도 하나의 문자 값 또는 A, B, ..., Z 이외의 47 미만의 문자 값이 후속하는 적어도 하나의 시프트 문자, 예를 들면 코드 93에 대해 정의되지 않고, 93i 종료 문자로 종료되는 문자 스트링을 포함한다.

제3 유효 기호 타입은 93i 시작 문자로 시작하고, 93i 심볼로지에 따른 임의의 데이터 문자 조합을 나타내는 기호 문자들의 중앙에 특수 형상의 프래그(아래 기재함)를 포함하고, 코드 93 종료 문자로 종료한다. 제3 유효 기호는 93i 에러 정정된 기호이다. 제4 유형의 유효 기호는 93i 시작 및 종료 문자로 시작 및 종료되고, 임의의 데이터 문자 조합을 포함하지만, 이러한 시작/종료 문자 조합은 짝 기호가 존재하는 것을 지시한다. 예를 들면, 93i 시작 및 종료 문자들을 갖는 93i 기호는 라인 주사 가능한 기호가 93i 기호에 근접하게 또는 그 근처에 위치하는 것을 판독기에 나타내고, 전형적으로 93i 기호 내에 인코딩된 데이터에 연관된 인코딩된 데이터를 포함한다. 짝 기호의 사용은 아래 보다 상세히 기재한다.

하기 테이블은 93i 심볼로지에 따라 시작/종료 문자 조합을 요약한다. 하기 테이블에서 "C93"은 코드 93 시작/종료 문자를 의미하는 한편, "EC"는 에러 정정을 나타낸다.

시작/종료 문자	유형	구별되는 특징
C93/C93	코드 93	(46보다 큰 값의 문자 또는 불법적 시프트 조합 없음)
C93/93i	93i	(93i 종료 문자 및 적어도 하나의 기타 93i 특수 특징을 가짐)
93i/C93	EC에 따른 93i	(93i 시작 문자 및 항상 52 값의 문자를 가짐)
93i/93i	비교에 따른 93i	(시작 및 종료 문자들로서 적어도 2개의 93i 문자들을 가짐)

인접한 태그 지시 특징(아래 기재함)은 93i 시작/종료 문자 조합중 어느 것에 의해 사용될 수 있음에 주의하자. 본 명세서에 명확하게 정의되지 않은 모든 문자 조합 또는 모드 값들은 무효이며, 디코딩 오퍼레이션을 실패하게 만든다.

코드 93 심볼로지에서의 값들을 넘는 문자 값들을 갖는 문자들 뿐만 아니라, 93i 심볼로지에 따른 독특한 시작 및 종료 문자들을 사용함으로써, 적어도 2가지 패턴 변화는 표준 코드 93 기호에 따라 93i 기호를 혼동시키는 데 필요하다. 예를 들면, 93i 시작 또는 종료 문자, 및 46보다 큰 값을 갖는 문자는 코드 93 시작/종료 문자 및 46보다 작은 값을 갖는 문자로서 해석되어야 한다. 부가적으로, 체크 문자는 표준 코드 93 기호로서 해석될 93i 기호에 대해 전형적으로 오역될른지 모른다.

코드 93 심볼로지과 달리, 93i 심볼로지는 코드 93 심볼로지에 사용된 47보다는 53 기호 값을 사용한다. 보다 상세하게는, 93i 심볼로지는 기호값 47-52 및 2개의 시작 문자들을 부가한다. 2,1,3,1,1,1의 바 스페이스 바 패턴으로 시작하는 제1 시작 문자는 기호가 93i 기호임을 나타내고, 따라서 기호 값 47-52를 포함할 수 있다. 93i 시작 문자에서 단일 폭 스페이스에 의해 분리된 2개의 넓은 바들을 사용함으로써, 판독기는 단일 폭 스페이스가 분해되었는지 여부를 결정함으로써 기호가 또렷한지 여부를 결정할 수 있다. 판독기가 93i 시작 문자에서 5-폭의 바를 식별하는 경우, 판독기는 1-넓이 스페이스 분해되지 않기 때문에 기호마 마찬가지로 흐릿함을 판단한다.

다른 시작 문자는 코드 93 심볼로지에 대한 시작 기호와 동일하고, 기호가 유일한 유효 코드 93

기호 문자들을 사용하여 인코딩되는 것을 지시한다. 고르게 혼합된 영숫자 메시지에 대해, 상위 케이스 영문자만을 사용함으로써, 코드 93 심볼로지는 가장 효율적인 문자 밀도를 제공하고, 따라서 그러한 메시지에 대해 바람직하다.

도 3에 나타낸 바와 같이, 93i 심볼로지에서 각각의 기호 문자에 대한 기호 문자 구조는 9개의 모듈 내에 3개의 바 및 3개의 스페이스들을 사용한다. 각각의 바 또는 스페이스는 1, 2, 3 또는 4 모듈 폭이다. 코드 93 심볼로지에서와 같이, 93i 심볼로지는 X 치수의 10배와 동일한 최소 폭을 갖는 리딩 안정 존(QZ), 2개의 시작 문자들중의 하나, 데이터 문자들을 인코딩하는 1개 이상의 기호 문자, 2개의 기호 체크 문자("C" 및 "K"라 칭함), 종료 기호 문자 및 꼬리 안정 존을 사용한다. 기호는 이하 고찰되는 바와 같이 기호의 중앙에 에러 정정 또는 특수 기능 플래그 및 메시지 길이 지시기, 및 5개의 에러 정정 문자를 포함할 수 있다. 도 4는 각각의 기호 문자에 대해 대응하는 데이터 문자와 함께, 각각의 기호 문자 간의 짧은 수직 라인으로 나타내는 개개의 기호 문자들로 분석되는 라벨 101(에러 정정 없음)의 기호 문자들을 보여준다.

상기한 바와 같이, 각각의 93i 기호는 종료 기호 문자 직전에 2개의 체크 문자들을 포함한다. 모듈로 53 합은 기호 내의 모든 기호 값에 대한 체크 알고리즘에 따라 사용된다.

코드 93 심볼로지에 의해서와 같이, 체크 문자 "C"는 도 2에 나타낸 바와 같이 가중 시퀀스에 의해 배가된 기호 값들의 곱의 모듈로 합에 기초하여 연산된다. 직전에 선행하는 문자로 시작하는 우측으로부터 좌측으로의(종료 기호 문자로부터 시작 기호 문자로의) 가중 시퀀스는 반복되는 시퀀스 1,2,3,...20,1,2,3,...20, 1,2,...이다. 체크 문자 "K"는 기호 값들과 상이한 가중 시퀀스의 곱의 모듈로 합에 기초하여 생성되고, 여기서, 체크 문자 "C"로 시작하여 우측으로부터 좌측으로의 가중은 반복되는 시퀀스 1,2,3,...15,1,2,3,...15,1,2,...이다. 코드 93 심볼로지에서와 같이, 시작 및 종료 기호 문자들은 체크 문자 산출에 포함되지 않는다.

93i 기호 위에 위치된 짝 PDF417 기호와 함께 93i 기호를 보여주며, 예를 들면, E 4의 기호를 고려하면, 데이터 문자들은 좌측에서 우측으로 9,3,i,[ECI 16], {30908}이고, "[ECI 16]"은 ECI 값 000016을 의미하는 한편, "{30908}"은 유니코드 표준에서 16-비트 코드 30908을 갖는 아시아 문자이다(대략적으로 "MA"로 표명됨). 본 명세서에 일반적으로 사용된 바와 같이, 단독으로 사용된 "기호"라는 용어는 라벨 101에 나타낸 것 등의 기호 문자들의 수집을 의미한다. 숫자 데이터 문자 "9" 및 "3"은 직접적으로 인코딩되는 한편, 데이터 문자 "i"는 시프트 문자에 의해 (아래 기재되는 바와 같이) 인코딩되어야 한다. 본 명세서에 일반적으로 사용되는 바와 같이, 단독으로 사용된 "문자"라는 용어는 데이터 문자 또는 그의 대응하는 기호 값을 의미한다. [ECI 16] 데이터 문자는 2개의 기호 문자들로 형성되는 한편, 아시아 문자(30908)는 아래 고찰되는 워드 모드를 사용한다. 간단히 말하자면, 30908값은 식 $(16*43^2)+(30*43)+34$에 따라 인코딩된다. 결과적으로, 데이터를 인코딩하기 위한 기호 문자들의 생성된 스트링에 대한 기호 값들은 [09][03][46][18][47][16][50][16][30][34]이다. 적절히 가중된 상기 체크 문자 알고리즘을 사용함으로써, 우측으로부터 좌측으로 체크 문자 "C"에 대한 연산은 다음과 같다:

"C" = (10x9 + 9x3 + 8x46 + 7x18 + 6x47 + 5x16 + 4x50 + 3x16 + 2x30 + 34) 모드 53

= 1315 모드 53

= 43

마찬가지로, 체크 문자 "K"는 "C"를 포함하는 가중치로부터 산출된다:

"K" = (11x9 + 10x3 + 9x46 + 8x18 + 7x47 + 6x16 + 5x50 + 4x16 + 3x30 + 2x34 + 43) 모드 53

= 1627 모드 53

= 37

상기한 바와 같이, 93i 심볼로지는 여러 가지 특수 문자들을 사용한다. 코드 93 심볼로지에 의해서와 같이, 93i 심볼로지는 도 2에 나타낸 기호 값 43-46을 갖는 4개의 시프트 문자들 [S1]-[S4]를 사용한다. 기호 값 10-35에 선행하는 시프트 문자는 도 5의 테이블에 나타낸 바와 같이 단일의 완전한 ASCII 데이터 문자를 나타낸다. 도 5의 제2 칼럼에서 [S3]A 내지 [S3]ZDML 문자 조합이 유효하고, 지시된 단일 문자들과 연관된 ASCII 문자들을 생성하기 위해 93i 심볼로지에 따라 사용될 수 있다. 예를 들면, 데이터 문자 "Q"는 단일 기호 값 [81] 또는 2개의 기호 값 [S3][81]로 나타낼 수 있다. X, Y 또는 Z를 갖는 문자 쌍들 [S2]는 모두 ASCII 값 DEL(삭제)을 인코딩한다.

시프트 문자 [S1]-[S4]는 0 내지 9 또는 36 내지 46의 기호 값이 후속할 때, 3 내지 23개의 시프트된 문자의 연속적인 스트링 또는 그룹이 특정 초기 시프트 문자에 기초하여 지시된다. 도 5에 나타낸 바와 같이, 문자 스페이스(sp), $, %, +, -, ., / 및 숫자 디지트 0-9는 시프트되지 않으며, 어떠한 시프트된 문자를 제공하기 위해 사용되지 않는다. 따라서, 문자 값들 0-9 및 35-46을 갖는 이러한 문자들은 추가의 기능성, 즉 그룹 시프트 기능을 제공하기 위해 사용될 수 있다. 그룹에서 시프트된 문자들의 수는 하기 테이블에 기초하여 지시된다:

[표 1]

시프트 문자 및 후속 문자 값	후속하여 시프트된 문자들의 수(그룹)
[S?][0]	3
[S?][4]	7
[S?][8]	11
[S?][38]	15
[S?][42]	19
[S?][46]	23
[S?][1]	4
[S?][5]	8
[S?][9]	12
[S?][39]	16
[S?][43]	20
[S?][2]	5
[S?][6]	9
[S?][36]	13
[S?][40]	17
[S?][44]	21
[S?][3]	6
[S?][7]	10
[S?][37]	14
[S?][41]	18
[S?][45]	22

여기서 "[S?]"는 시프트 문자들 [S1]-[S4] 중의 하나에 대응한다. 달리 말하자면, 기호 값 0-9 또는 36-46이 후속하는 4개의 시프트 문자들 [S1]-[S4]중의 하나는 소정의 수의 후속 문자들에 대한 래치 기능을 제공하고, 후속 문자들은 초기 시프트 문자에 기초한 이들의 대응하는 시프트된 값으로 래치된다. 예를 들면, 데이터 "모듈로 53"이 인코딩되는 경우, 기호 값들 "22, 46, 02, 24, 13, 30, 21, 24, 38, 05 및 03"에 대응하는 93i 문자들 "M[S4][S2]모듈로[sp]53"이 사용된다.

단일 시프트 문자는 (1) 선행하는 시프트 문자에 기초한 하나의 후속 문자를 시프트시키거나, 또는 (2) 문자들이 초기의 시프트 문자에 의해 지시된 동일한 세트로 현재 시프트되는 경우, 하나의 후속 문자를 베이스 93i 문자 세트로 시프트시키기 위해 시프트된 문자들의 그룹 내에서 사용될 수 있다. 시프트된 문자들 및 하나의 후속하는 문자의 스트링 내에서 모든 시프트 문자는 표 1의 그룹 길이에 포함된다. "시프트된 스트링" 및 "시프트된 문자"는 도 5의 데이블에 기초한 시프트 문자들 [S1]-[S4] 중의 하나를 사용하여 유도된 데이터 문자들의 스트링 또는 개개의 데이터 문자를 의미한다.

예를 들면, 데이터 메시지 "모듈로 53 매쓰"가 인코딩되는 경우, 다음 93i 문자들이 사용된다:

"M[S4][37]ODULO[sp]53[sp][S4]MATH"

문자 값 [37]은 상기 표 1에 기초하여 하기 14개의 문자들이 도 5의 표에 따라 시프트되는 것에 주의하자. 시프트 문자 [S4]는 14개의 시프트된 문자들의 스트링에 포함되기 때문에, 그룹에서 직후에 후속하는 문자는 시프트되지 않거나, 또는 데이터 문자들의 베이스 93i 세트에 대응한다. 시프트 문자 [S1]-[S3]이 대신에 사용된 경우, 직후에 후속하는 문자는 3개의 시프트 문자들이 사용된 것에 기초하여 도 5의 표에 따라 시프트된다. 문자 값 0-9(데이터 문자들 0-9에 대응함) 또는 문자 값들 36-42(데이터 문자들 ", ., sp, $, /, +, %)이 시프트된 문자들로 둘러싸인 기호에 나타나는 경우에도, 그러한 문자 값들은 시프트된 스트링의 일부로서 직접적으로 인코딩되는 것에 주의하자. 숫자 모드 또는 기능 1(FNC1) 문자 (아래 논의함)는 시프트된 스트링 내에서 사용될 수 있지만, ECI, 바이트 모드 및 워드 모드 문자(아래 논의함)는 사용될 수 없다.

93i 심볼로지는 기호 값 47을 갖는 ECI 문자 [47]을 사용하고, 주어진 기호에서 규정된 의미의 바이트 또는 후속 데이터에 관한 정보를 인코딩한다. AIMI ECI 할당 서류는 ECI 수치 및 이 ECI 수치에 기초한 바이트 또는 데이터의 의미를 할당한다. ECI 수치는 000000 내지 811799 범위이다. 예를 들면, 하나의 ECI 번호는 국제적인 문자 세트의 인코딩을 나타낸다. 93i 심볼로지는 기호 내의 어느 곳에 ECI 번호를 위치하고, 이를 도 2의 기호 값 0-51로부터 선택된 1,2,3 또는 4 기호 값으로 후속시킴으로써 ECI 번호를 인코딩한다.

ASCII 값 92(도 5 참조)를 갖는 역슬래쉬 문자 "\"(역 사선)은 6개의 디지트 ECI 값 전에 전송된다. 역슬래쉬 문자는 기호가 판독될 때 생성된 기호 값들 또는 데이터의 스트링을 수신하는 호스트 컴퓨터 또는 시스템에 대한 에스케이프 문자로서 작용한다. 역슬래쉬 문자가 인코딩된 데이터 내에 놓여야 하는 경우, 2개의 역슬래쉬 문자들은 그 기호 내에서 인코딩되어야 함으로써 호스트는 ECI 값보다는 오히려 단일 역슬래쉬 문자가 바람직함을 알게 된다. 마찬가지로, 2개의 역슬래쉬 문자가 바람직한 경우, 4개의 역슬래쉬 문자가 인코딩되어야 호스트가 2개의 역슬래쉬 문자가 바람직함을 알게 된다. 93i 심볼로지에 따른 ECI 수치의 인코딩 규칙은 아래 표 2에 나타낸다.

요약하자면, ECI 수치 0-899에 대해, 이러한 수치에 후속하는 바이트 또는 데이터는 데이터가 다른 방식으로 압축될 수 있는 경우에 직접적으로 인코딩된다. 예를 들면, ECI 번호 89는 인코딩된 데이터의 특정 유형의 시작을 나타낼 수 있다. 이후에 후속하는 인코딩된 데이터는 압축되면서 직접적으로 인코딩된다. 그러나, 디지트 또는 완전한 ASCII 문자들이 0-899 간의 ECI 번호 후에 인코딩되는 경우, 숫자 모드 또는 바이트 모드(아래 기재됨)는 도 5에 나타낸 바와 같이 하위 128 ASCII 값 및 대응하는 기호 값을 사용하기 위해 사용될 수 있다. ECI 번호가 워드 모드(아래 기재됨)에 따라 인코딩된 데이터 문자들의 스트링 내에 인코딩되는 경우, 그에 매달린 8개의 제로를 갖는 완전한 128 ASCII 값이 사용된다. ECI 번호 900-811799는 이들의 가장 효율적인 모드의 바이트로서 인코딩되고, 워드 모드 문자([50])는 금지된다. 예를 들면, ECI 번호 950에 대해, 93i 기호 값 0-9의 스트링이 인코딩될 필요가 있는 경우, 데이터 문자들의 값이 ECI 950에 의해 특정된 모드에 따라 사용된 수치에 대응하지 않는 경우조차, 숫자 모드가 사용된다.

아래 표 2는 93i 심볼로지에 따라 ECI 값들을 인코딩하기 위한 규칙을 요약한다. 아래 표 2에서, "div"는 정부 분할 연산자를 의미하는 한편 "mod"는 모듈로 분할 연산자를 의미한다. "C1"은 가장 중요한 위치를 의미하는 한편, "C4"는 가장 중요치 않은 위치를 의미한다.

[표 2]

ECI 값	문자	값	범위
000000- 43	C1	ECI_val	C1=0 내지 43
000044- 95	C1 C2	44 ECI_val-44	C1=44 C2=0 내지 51
000096- 2799	C1 C2 C3	45 (ECI_val-96)div 52 (ECI_val-96)mod 52	C1=45 C2=0 내지 51 C3=0 내지 51
002800- 811799	C1 C2 C3 C4	((ECI_val-2800) div 140608)+46 ((ECI_val-2800)div 2704)mod 53 ((ECI_val-2800)div 52) mod 52 (ECI_val-2800)mod 52	C1=46 내지 51 C2=0 내지51 C3=0 내지 51 C4=0 내지 51

예를 들면, 000020의 ECI 값을 인코딩하기 위해, 하기 2개의 문자 스트링들이 사용된다: [47][20] 여기서 [47]은 ECI 기호 값 47이고, [20]은 93i 문자 "K"이다.

002000의 ECI 값을 인코딩하기 위해, 하기 단계들이 3가지 요구되는 문자를 얻는데 후속한다:

[47][45][(ECI_val-96)div 52][(ECI_val-96)mod 52]=

[47][45][1904 div 52][1904 mod 52]=

[47][45][36][32]

마지막으로, 200000의 ECI 값을 인코딩하기 위해, 하기 단계들이 4가지 요구되는 문자를 얻는데 후속한다:

[47][(ECI_val-2801)div 140808+46][(ECI_val-2801)div 2704)mod 52] [(ECI_val-2801)div 52)mod 52][(ECI_val-2801)mod 52]=

[47][197199 div 140808 + 46][197199 div 2704 mod 52] [197199 div 52 mod 52][197199 mod 52]=

[47][1+46][72 mod 52][3792 mod 52][15]=

[47][47][20][48][15]

기호 값 48은 본 명세서에서 "숫자 모드"라 칭하는 코드 93 심볼로지에서 숫자 압축 모들 나타낸다. 숫자 모드에 따라, 5개의 숫자 디지트는 3개의 기호 문자들로 압축된다. 따라서, 5개 이상의 디지트의 시퀀스는 5/3 숫자 모드를 사용하여 압축되어야 한다. 숫자 모드 문자, 기호 값 48은 5/3 숫자 압축 모드로 및 그로부터 벗어나 토글된다. 마찬가지로, 바이트 모드 및 워드 모드 문자들인 기호 값들 49 및 50 각각은 숫자 모드로부터 벗어나도록 사용될 수 있다. 기호가 숫자 모드에서 종료되는 경우, 숫자 모드 배출 문자는 불필요하다.

숫자 모드에 따라 5개의 숫자 디지트는 3개의 기호 문자들로 나타내고, 여기서 기호 문자들 각각은 0-47 범위의 기호 값을 갖는다. 5개의 디지트 숫자 스트링은 하기 수학식에 의해 생성된다.

$$A * 48^2 + B * 48 + C$$

여기서, A, B 및 C는 93i 기호 값이다. 5보다 크지만 5의 정확한 배수는 아닌 디지트의 스트링들이 인코딩될 때, 하기 4가지 규칙이 적용된다. 첫째, 스트링에서 5의 배수보다 1개 더 많은 디지트가 단일 기호 문자(기호 값 00-09)에 의해 직접적으로 인코딩된다. 둘째, 숫자 스트링이 5의 배수보다 2개 더 많은 디지트를 포함하는 경우, 마지막 7개의 디지트들은 아래 제3 및 제4 규칙에 의해 기재된 바대로 각각 나타내는 바와 같이 3개의 디지트 세트가 후속하는 4개의 디지트 세트로 분리된다. 셋째, 숫자 스트링이 5의 배수보다 3개 더 큰 디지트를 포함하는 경우, 스트링 말단의 3개의 디지트는 하기 수학식에 따라 2개의 기호 문자들로 나타낸다.

$$48 * A + B$$

다시, 여기서, A 및 B는 93i 기호 값이다. 넷째, 숫자 스트링이 5의 배수보다 4개 이상의 디지트를 포함하는 경우, 마지막 4개의 디지트는 상기 수학식(1)에 따라 3개의 기호 문자들로 인코딩되고, 여기서 수학식(1)에 따라 생성된 값은 100,000과 109,999 사이에 있다. 하기 표 3은 12345 내지 123456789 범위의 전형적인 5,6,7,8,9 및 10 디지트 스트링을 보여주고, 생성된 최적의 기호 값들은 숫자 모드에 따라 결정되었다.

[표 3]

전형적인 데이터	최적의 생성된 기호 값들
12345	[05][17][09]
123456	[05][17][09][06]
1234567	[43][45][02][11][39]
12345678	[05][17][09][14][06]
123456789	[05][17][09][46][16][37]
1234567890	[05][17][09][29][22][18]

ECI 값은 숫자 모드의 일부로서 사용되고, 여기서 기호 값 [47]은 ECI 프로토콜을 야기하지 않고, 대신에 5/3 숫자 압축 방법으로 자체 사용된다. 숫자 모드는 ECI 값이 사용되기 전에 먼저 배출되어야 한다.

기호 값 49는 본 명세서에서 "바이트 모드"라 칭하는 93i 심볼로지에서 바이트 모드를 나타낸다. 바이트 모드에 따라, 93i 심볼로지는 완전한 또는 확장된 ASCII 데이터 또는 스트레이트 바이트 데이터의 스트링들을 효율적으로 인코딩한다. "바이트"는 전형적으로 8-비트 세트의 데이터를 의미한다. 하기 수학식 3에 따라, 더블-바이트, 또는 2개의 8-비트 바이트가 3개의 기호 문자들 각각에 대해 인코딩된다:

$$A * 43^2 + B * 43 + C$$

여기서, A, B 및 C는 0과 42 사이의 93i 기호 값이다.

수학식 3에 따라 0과 65,535 사이의 조합된 값을 갖는 2개의 바이트는 2개의 기호 문자들로서 인코딩된다(즉, 2^{16}=65,536). 65,536 내지 75,535 사이의 수학식 3으로부터 초래되는 값들은 4개의 디지트를 인코딩하는 한편, 75,535 내지 76,535,의 값은 3개의 디지트를 인코딩한다. 결과적으로, 바이트 모드는 바이트 모드에서 숫자 문자들의 스트링을 인코딩하기 위한 정보 밀도를 개선시키기 위해 3- 및 4-디지트 숫자 간결화 방법을 제공한다. 76,536 내지 79,506 사이의 수학식 3으로부터 초래되는 값들은 정의되지 않으며, 판독기가 에러 신호를 디코딩하고 출력하는 데 실패하게 만든다.

숫자 모드에 의해서와 같이, 바이트 모드가 도입되고, 기호 값 49를 갖는 바이트 모드 기호 문자를 사용하여 배출된다. 바이트 모드는 워드 모드 기호 문자(기호 값 50) 또는 숫자 모드 기호 문자(기호 값 48)를 사용함으로써 역시 배출될 수 있다. 부가적으로, 기호 값 43-46을 갖는 시프트 문자 [S1]-[S4]는 판독기가 바이트 모드를 빠져나가게 하고, 다음 문자들에 대한 기호 값들에 128을 부가하게 한다. 따라서, 확장된 ASCII 문자들은 바이트 모드를 빠져나간 후 효율적으로 인코딩될 수 있다. 마찬가지로, 하나의 문자가 기호의 말단에 남겨지면서, 선행하는 문자들이 바이트 모드에 있게 되는 경우, 최종 문자는 그의 기호값+128에서 디코딩된다(확장된 ASCII에서와 마찬가지). 기호가 바이트 모드로 유지되면서 종료되는 경우, 최종 바이트 모드 기호 문자(기호 값 49) 등의 배출 모드 문자는 불필요하다. 2개의 문자들이 기호의 말단에 남겨지는 경우, 2개의 문자들은 바이트 모드가 퇴진된 경우와 같이 이들의 베이스 기호 값에서 디코딩된다.

기호 문자 인코딩 효율을 개선시키기 위한 여러 가지 인코딩 전략은 바이트 모드에서 93i 심볼로지에 따라 허용된다. 예를 들면, 바이트 모드는 2개의 확장된 ASCII 데이터 문자들이 3개의 기호 문자들의 단일 그룹으로서 인코딩될 수 있게 한다. 확장된 ASCII 데이터 문자에 의해 종료되는 완전한 또는 확장된 ASCII 데이터 문자들중 짝수에 대해, 3개의 기호 문자들의 그룹이 93i 심볼로지에서 바이트 모드에 따라 사용된다. 확장된 ASCII 데이터 문자에 의해 종료되는 혼성의 완전한 확장된 ASCII 데이터 문자의 홀수에 대해, 짝수 문자들은 3개의 기호 문자들로 나타내고, 최종(또는 유일한) 데이터 문자가 2가지 방식중 하나로 인코딩된다. 첫째, 최종 기호 문자가 확장된 ASCII 문자인 경우, 이는 도 5의 적절한 시프트 문자로 처리된 완전한 ASCII 문자로서 인코딩된다. 둘째, 최종 문자가 단일의 완전한 ASCII 문자인 경우, 이는 직접적으로 인코딩되고, 바이트

모드 문자가 후속한다. 두 경우, 최종 문자는 128의 기호 값 + 베이스 문자 또는 시프트 문자의 값을 갖는다(도 2에 나타낸 바와 같이 기호 값 00-46).

하기 표 4는 바이트 모드에 따른 여러 가지 스트링의 데이터 문자들의 최적 인코딩을 제공한다. 상기하면, 기호 값 [49]는 바이트 모드 문자를 의미하는 한편, [S?]는 43-46의 기호 값들을 갖는 4가지 시프트 문자들 중의 하나를 나타낸다. 하기 표 4의 3번째 칼럼에서, 문자 "A" 내지 "F"는 도 2의 기호 값 00-42를 갖는 임의의 데이터 문자를 의미한다.

[표 4]

문자 번호	데이터 문자 유형	93i 문자 스트링
1 1 1	표준 완전한 ASCII 확장된 ASCII	A [S?]A 값에 좌우되어 [43][S?]A 또는 [49]A[49]
2 3	확장된 ASCII-이어서 퇴진 완전/확장되고 이어서 확장된-퇴진	[49]ABC[49] [49]ABC[S?]A 또는 [49]ABC A[49]
4 5	" "	[49]ABC DEF[49] [49]ABC DEF[S?]A 또는 [49]ABC DEF A[49]
. . .	. . .	. . .

숫자 데이터 문자들의 스트링이 완전한 또는 확장된 ASCII 데이터 문자들 사이에서 인코딩되어야 할 때, 부가적인 인코딩 전략은 기호 문자 밀도를 개선시키기 위해 93i 심볼로지에 따라 이용될 수 있다. 하나 또는 2개의 숫자 데이터 문자들이 완전한 또는 확장된 ASCII 문자들 사이에서 인코딩되는 경우, 1개 또는 2개의 숫자 문자들은 디지트의 수에 따라 도 5에 나타낸 바와 같이 ASCII 값 48 내지 57을 갖는 단일의 완전한 ASCII 문자들로서 처리된다. 3개 내지 9개의 숫자 데이터 문자들이 완전한 및 확장된 ASCII 문자들 사이에서 인코딩되는 경우, 3개 및 4개의 디지트의 그룹은 바이트 모드의 값 65536-75535 및 75536-76535에 따라 압축된다. 달리 말하자면, 바이트 모드는 완전한 또는 확장된 ASCII 문자들의 스트링 내에서 3 내지 9개의 디지트 숫자 스트링에 대한 만족스러운 숫자 압축을 제공한다. 그러나, 10개 이상의 디지트의 스트링에 의해, 바이트 모드는 퇴진되어야 하고, 숫자 모드는 완전한 또는 확장된 ASCII 문자들의 스트링 내에 숫자 모드 문자 [48]을 간단히 제공함으로써 도입된다.

1개, 2개, 3개 또는 4개의 표준 또는 베이스 93i 데이터 문자들이 완전한 또는 확장된 ASCII 문자들 내에서 인코딩되어야 하는 경우, 그러한 베이스 데이터 문자들은 바이트로서 처리된다. 그러나, 5개 이상의 베이스 데이터 문자들이 완전한 또는 확장된 ASCII 문자들의 중앙에서 인코딩되어야 하는 경우, 바이트 모드 문자 [49]를 먼저 인코딩함으로써 바이트 모드에서 벗어나 시프트하고, 직접적으로 5개 이상의 베이스 데이터 문자들을 인코딩하고, 이어서 다른 바이트 모드 문자 [49]에 의해 바이트 모드로 재진입하는 데 보다 효과적이다. 하기 표 5는 완전한 또는 확장된 ASCII 문자들의 스트링 내에 놓인 디지트의 스트링들의 예를 제공한다. 하기 표 5에서, 문자 유형들은 하이픈으로 분리된다.

[표 5]