

(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(51) Int. Cl.⁶
H04N 7/24

(11) 공개번호 특 1998-042030
(43) 공개일자 1998년 08월 17일

(21) 출원번호	특 1997-057601
(22) 출원일자	1997년 11월 01일
(30) 우선권주장	60/030,108 1996년 11월 01일 미국(US)
(71) 출원인	텍사스 인스트루먼츠 인코포레이티드 윌리엄 비. 캠플러
(72) 발명자	미국 텍사스주 달拉斯 노스 센트럴 익스프레스웨이 13500 샤우밸 게라드 프랑스 06600 안티베스 체 드 라 슈게트 300 레스 베르게스 드 발 콘스탄스 라세르 세르게 프랑스 83600 프레주스 세인트 진 드 카네스 루 드 메이르사오우 29 지아니 마리오 프랑스 06800 카그네스 서 메르 루트 드 그라쎄 73 알이에스. 르오아시스-비 에이티. 비 스피츠 티에멘 네덜란드 노스 리치랜드 힐스 엘리스 드라이브 8528 장수길, 주성민
(74) 대리인	장수길, 주성민

심사청구 : 없음

(54) MPEG2 트랜스포트 스트림 패킷 파서 시스템

요약

본 발명은 관련 플래그로 사전 처리한 후의 패킷을 갖는 중간 버퍼, 및 송신 플래그에 의해 선택된 패킷의 추가 소프트웨어 처리를 위한 프로세서를 포함하는 트랜스포트 스트림 패킷 파서 시스템(transport stream parser system)에 관한 것이다.

대표도

도5

명세서

도면의 간단한 설명

- 도 1은 188 바이트를 갖는 트랜스포트 패킷을 도시한 도면.
- 도 2는 종래의 디코더 시스템을 도시한 도면.
- 도 3은 데이터 패킷을 인입하기 위한 두 데이터 흐름 경로를 도시한 도면.
- 도 4는 인터럽트를 발생시키기 위한 정보 및 조건을 도시한 도면.
- 도 5는 본 발명의 시스템에 대한 블록도.
- 도 6은 패킷을 분석하는데 사용되는 흐름도.
- 도 7은 PES 패킷의 구성을 설명하는 도면.

발명의 상세한 설명

발명의 목적

발명이 속하는 기술분야 및 그 분야의 종래기술

본 발명은 MPEG2(Moving Picture Experts Group: 동화상 표준화 그룹 2) 디코더에 관한 것으로, 특히, 이러한 디코더에 사용되는 MPEG2 트랜스포트 스트림 패킷 파서(transport stream packet parser)에 관한 것이다.

디지털 TV 디코더는 다음의 5개 모듈로 분할될 수 있다.

복조

MPEG 트랜스포트 패킷 파서(TPP)

MPEG 오디오

MPEG 비디오

RISC(Reduced Instruction Set Computer) 마이크로-컨트롤러.

MPEG2 트랜스포트 스트림 파서(TPP)는 MPEG 트랜스포트 스트림을 수신하고 비디오, 오디오 또는 서비스 정보 패킷을 선택한다. 디코딩 후, 패킷은 데이터 스트림을 형성하기 위해 메모리 버퍼에 저장된다. 오디오 디코더는 MPEG 오디오 스트림을 처리하여 아날로그 오디오 신호를 생성한다. 비디오 디코더는 MPEG 비디오를 감압시켜 비디오 시퀀스를 발생시킨다.

트랜스포트 패킷 포맷

트랜스포트 패킷(도 1)은 188바이트를 포함한다. 패킷은 페이로드(payload)의 32비트 헤더 및 184 바이트를 두 부분으로 분할한다. 트랜스포트 헤더는 트랜스포트 패킷 파서가 코오스 필터(coarse filter) 처리할 수 있게 하는 상이한 필드(PID, 페이로드 유닛 시작 지시자, 적응 필드 플래그, 연속성 카운터 인덱스)를 포함한다. 트랜스포트 헤더 다음에, 패킷은 가변 길이의 적응 필드 또는 페이로드를 포함할 수 있다. 이들 페이로드 종의 일부는 가변 길이를 갖는 헤더(PES 헤더)에서 자체 시작하게 될 것이다(도 1 참조). MPEG2 표준의 PES 헤더는 패킷보다 더 큰 것이 가능하다(그 다음 패킷에 대한 오버플로우).

MPEG2 트랜스포트 스트림 시스템 분석기에 대한 기술에는 완성된 트랜스포트 스트림 하드웨어 분석기 및 마이크로 프로세서를 포함한 칩 세트 해석으로 구성되는 것으로 진술된다. 비디오 및 오디오 디코더는 도 2에 도시된 바와 같이 분리 디바이스 내에 있다. 트랜스포트 스트림 파서는 현재의 접근법으로는 데이터를 올바른 목적지(right destination)로 진행하게 하는 업무를 CPU에 남기고 상이한 모듈을 제어하는 트랜스포트 스트림 풀 분석(약 70k 게이트)을 하고 있다. 다른 해결책으로는 모든 트랜스포트 분석을 소프트웨어로 실행하는 것이다. 이는 더 높은 CPU 전력 처리를 요구할 것이다.

발명이 이루고자 하는 기술적 과제

트랜스포트 스트림 파서 시스템은

관련 플래그로 사전 처리(pre-processing)한 후의 패킷을 포함한 중간 버퍼(intermediate buffer), 및 송신 플래그에 의해 선택된 패킷의 추가 소프트웨어 처리를 위한 프로세서를 포함한다.

발명의 구성 및 작용

본 발명의 트랜스포트 스트림 파서는 더욱 효과적으로 분석하기 위해 하드웨어 및 소프트웨어 간의 트랜스포트 스트림 및 MPEG2 분석 로드를 분리한다. 하드웨어 필터링은 PIDs 인식에 기초하여 조기 판정(early decision)을 얻을 수 있게 하며, 펌웨어(firmware)가 데이터 스트림을 매우 효과적으로 감소시키는 반면, 감소된 데이터 스트림에 대해 더욱 복잡한 필터링 알고리즘을 실행한다. 이러한 접근법으로 자원을 최적화시킨다. 수신하는 비트율이 펌웨어에 의해 사전 처리되기 때문에 펌웨어가 더 낮은 CPU 전력을 요구하더라도 하드웨어는 간단하게 유지된다.

패킷의 파싱(parsing) 및 루팅은 각 PIDs에 대한 로컬 메모리에 저장된 속성 세트에 결합된 PID 인식 메카니즘으로 행해진다. 이들 속성은 상이한 헤더들로부터 수신된 정보를 완성한다.

프랜스포트 패킷 파서 기능

이 모듈의 기능은 응용에 의해 선택되지 않은 모든 패킷을 동시에 폐기시켜, 내장형(embedded) 소프트웨어로부터 실시간 도움없이 가능한 한 다수의 패킷으로서 루팅하는 것이다.

일단, 유효 PID를 가진 패킷이 검출되면, 파서는 패킷이 CPU에 의해 추가 처리를 요구하는지의 여부를 결정하는 조건 세트를 조사한다(EOP 인터럽트 발생에 대한 상세한 설명 참조). 이들 조건 중에서 하나가 만족되면, 패킷은 부착된 플래그를 가진 임시 버퍼에 송신되고(추가 처리가 요구됨), 인터럽트 EOP는 새로운 패킷이 수신되었음을 지시하기 위해 CPU에 송신된다. 그 후, 이런 임시 버퍼에 저장된 데이터는 도 3의 경로(2)에 도시된 바와 같이 루팅되거나 폐기될 수 있기 전에 내장형 소프트웨어에 의해 처리된다. 일단 데이터가 처리되면, RISC는 데이터를 자체에 전달하거나 시스템 버퍼로의 DMA 전달을 시작한다. 소프트웨어는 다음 패킷(동일 PID)의 루팅을 변경하기 위해 PID 속성의 일부(예로, 스트림-id: stream-id 필터)를 변경하는 경우도 있다. 어떤 조건도 만족하지 않을 경우, 부착된 플래그는 어떤 부가 처리도 요구되지 않음을 나타낸다. 패킷은 자동 DMA 전송(예로, 압축 비디오 정보를 포함한 패킷은 시스템 비디오 버퍼에 직접 송신될 수 있음)을 통하여 최종 목적지에 자동적으로 운반될 수 있다(경고 1). 이 경우, 어떤 EOP 인터럽트 RISC에 송신되지 않으며, 최종 목적지로의 패킷 전송은 어떠한 CPU 자원을 요구하지 않는다.

도 3은 상이한 두 데이터 흐름을 도시한다.

이러한 메카니즘(도 3)은 하드웨어/소프트웨어 분석 간의 교환(trade-off) 용키(key)이다. 트랜스포트 스트림 파서는 트랜스포트 스트림 소프트웨어 분석의 장에서 설명된 바와 같이 패킷을 매우 쉽게 루팅하여, 비트 스트림의 일부에 대한 더욱 복잡한 분석을 가능하게 한다.

이러한 선택은 헤더(PID, AF 플래그, PES/PSI 시작 플래그[페이로드 유닛 시작 지시자], 트랜스포트 에러 플래그, 카운터 연속성 인덱스), 및 PID 속성을 분석함으로써 이루어지는데, 왜냐하면, 항상 패킷 자체로부터 정보를 추출함으로써 선택될 수 있는 것이 아니기 때문이다. 때때로, 이전 패킷(동일 PID)의 속성은 패킷이 부가 처리(두 패킷 간의 PES 헤더 분리)를 요구하는지의 여부를 결정할 것이다. 때로는, PID

번호에 연결되기도 한다. PID가 항상 CPU 처리를 요구하는 순간에 테이블 정보를 포함하면, 속성(Spe1)이 설정된다(이 PID를 가진 패킷은 항상 CPU에 송신될 것임을 의미한다). EOP 인터럽트를 발생시키는데 사용되는 정보의 리스트는 도 4에 이하와 같이 요약되어 있다.

이 도면은 정보가 어디에서 유출되는지를 설명한다. EOP를 발생시키기 위한 조건은 이하 상세하게 설명되어 있다.

패킷 분석기 제어 레지스터 비트 설명

유효 PID(Valid PID): 이 비트는 PID 인식 기능의 결과이다. PID 번호는 각 패킷의 트랜스포트 스트림 헤더 내에 수신된 패킷 식별 번호이다. 그것은 패킷이 반드시 시스템으로 전송되는지의 여부를 결정한다. PID 번호가 선택될 경우, 유효 PID 비트는 이 PID 번호를 가진 패킷이 수신될 때 활성 상태 '1'이 된다. 그것은 패킷 필터링의 제 1레벨이고, 다른 분석 단계를 위한 필수 조건이다.

Te(트랜스포트 에러 지시자): 이는 트랜스포트 스트림 패킷 헤더로부터 검색된다. 트랜스포트 에러 지시자 비트는 수신된 패킷이 다소의 에러가 있을 때 활성상태에 있다.

Pes/Psi 시작 플래그(트랜스포트 스트림 페이로드 시작-유닛-지시자): 이는 트랜스포트 스트림 패킷 헤더로부터 검색된다.

트랜스포트 스트림의 페이로드의 경우, PES 패킷 데이터를 포함한다. '1'은 이 트랜스포트 스트림 패킷의 페이로드가 PES 패킷(PES 헤더)의 제 1바이트에서 시작하는 것을 나타내며, '0'은 어떤 PES 패킷도 이 트랜스포트 스트림 패킷 내에서 시작하지 않음을 나타낸다.

트랜스포트 스트림 패킷의 페이로드의 경우, PSI 데이터를 포함한다. '1'은 페이로드가 PSI 섹션의 적어도 1시작부를 운반하는 것을 나타내며, PSI 섹션의 어떤 시작도 운반되지 않을 경우의 '0'은 그 트랜스포트 스트림 패킷이다.

AF 플래그(트랜스포트 스트림 적용 필드): 트랜스포트 스트림 헤더의 2비트 필드는 트랜스포트 스트림 헤더 다음에 적용 필드가 따르는 지의 여부를 나타낸다.

AF 값	설명
00	ISO/IEC에 의해 미래용으로 보류
01	적용 필드가 없고, 페이로드만 존재
10	적용 필드만 있고, 페이로드는 없음
11	적용 필드 다음에 페이로드 존재

CC-에러: 이 비트는 연속 카운터(CC)에 대한 하드웨어 불연속성 카운터 검출 기능의 결과이다. 검출은 현재_CC라고 하는 CC(4비트는 트랜스포트 스트림 헤더에 포함됨)와 동일 PID의 이전 패킷으로부터의 CC(이전_CC라 함)와의 비교로 이루어진다. 이전_CC는 하드웨어에 의해 변경된 속성이다. 일단 비교되면, 하드웨어는 다음 패킷 등의 수신을 위해 현재_CC에 이전_CC를 덧쓴다. 그 비교 결과가 현재_CC = 이전_CC 또는 현재_CC = 이전_CC + 1을 일치시키지 않을 경우, CC_에러비트는 활성 상태가 된다.

복제-패킷 플래그(duplicate-packet flag): 비교 결과가 현재_CC = 이전_CC를 나타낼 경우, 복제-패킷은 활성 상태가 된다.

Hpei(패킷 인에이블 인터럽트의 하드웨어 엔드): 이는 하드웨어에 의해 설정된 속성 비트이다. [패킷 유닛 시작 지시자가 활성될 때와 프로그램 서비스 정보(PSI)를 가진 패킷이 아닐 때] 새로운 패킷의 수신시, Hpei는 활성 상태가 되고, 동일 PID의 다음 패킷에 대해 리셋된다. 이것은 PES 헤더가 두 패킷 사이에서 분리되는 경우에 PES 헤더를 가진 패킷 다음의 패킷이 추가 분석을 위해 임시 저장되게 한다. 이 비트는 인입 비트 스트림이 비트율이 EOP 서비스 루틴 레이턴시(latency) 이내에 고정하도록 2 하이(two high)일때 필요하다.

Spe1(패킷 인에이블 인터럽트의 소프트웨어 엔드): 이러한 속성 비트는 소프트웨어에 의해 설정되며, 상이한 두 가지 목적을 포함하고 있다. 제 1용법으로서, 비트는 항상 CPU 처리를 요구하는 PID(이를 테면, 테이블 정보를 갖는 PID)에 대한 초기화로 설정된다. 이러한 PID의 모든 패킷이 CPU에 의해 분석되게 하는 경우에 간단하게 사용된다.

제 2용법으로는 PES 헤더가 두 패킷 간에 분리되는 오디오 및 비디오 패킷에 대한 것이다. 이는, 일단 소프트웨어가 레이턴시 응답이 비트 스트림 풀에 필적된다고 판단할 경우, 소정의 시스템에 대해서만 타당할 뿐이다. 그 경우, 소프트웨어가 PES 헤더를 포함한 패킷의 EOP 인터럽트를 수신하자마자 Spe1를 설정한다. 그 후, PES 헤더가 다음 패킷에서 오버플로우하는지를 판단한다. 오버플로우하지 않을 경우에는 Spe1 속성 비트를 리셋한다. 이 경우, Spe1은 능률적으로 얻도록 디스에이블되는 Hpei를 대체할 수 있다(En_Hpei 참조).

PSI(프로그램 서비스 정보 패킷 플래그): MPEG2 트랜스포트 스트림에서, 트랜스포트 스트림 헤더에서의 1 플래그(페이로드 유닛 시작 지시자)는 새로운 PESD 패킷의 시작 또는 프로그램 서비스 정보 섹션의 시작을 나타낸다. 이를 두 종류의 데이터를 구별하기 위해서, 다른 속성이 요구된다(PSI). 이러한 속성은 프로그램 서비스 정보(PSI)를 포함한 PIDs에 대한 초기화로 펌웨어에 의해 설정된다.

이것은 Hpei를 발생시키는데 사용되지만, 암호화/클린(encrypted/clean) 바이트 간의 경계를 검출하는데 더욱 중요하다. 이 정보가 PIDs에 밀접하게 결합되지 않으면, 제어기는 암호화 플래그는 물론, 비트 스트림 인터페이스에 대한 다소의 입력 버퍼없이 정시에 암호화/클린 바이트 간의 경계도 발견할 수 없다.

스트림-id-필터: 이러한 소프트웨어 속성 비트는 선택되지 않은 스트림-id를 가진 PES 헤더를 포함한 패킷을 수신할 때 소프트웨어에 의해 설정된다. 이러한 속성이 일단 설정되면, 하드웨어는 새로운 PES 헤더에 도달할 때까지 페이로드(AF = 1x)만을 포함한 PID의 모든 패킷을 폐기시킨다. 패킷은 PES 헤더를 분석하는 CPU에 송신되고 새로운 스트림-id가 유효한지를 판정한다. 유효할 경우, 스트림-id-필터를 다음 패킷에 대해 리셋한다.

전체적인 제어 비트 설명

En Hpei: Hpei 메카니즘은 시스템의 제약(constraint)에 의존하여 오프/온으로 전환될 수 있다. 소프트웨어가 다음 패킷의 수신 전에 Spei 속성 비트를 재활성하여 변경시키기에 충분한 시간을 갖는 것으로 판단하면, Hpei 속성을 디스에이블할 수 있다. PES 헤더가 1 패킷 내에 전부 포함되지는 않을 때 추가 분석을 위해 CPU에 PES 헤더를 가진 패킷 다음의 패킷만이 송신되기 때문에 그 경우, 비트 스트림 파싱이 더욱 효과적이다.

En 에러 p: 전체적인 제어 비트는 임시 버퍼로의 에러 패킷(errorous packets)의 전송을 인에이블한다. 이는 주로 다소의 새로운 에러 잠복(errorous concealment)이 앞으로 발견될 수 있는 경우에 어느 정도 유연성을 부가하는 것이다.

En 복제 p: 전체적인 제어 비트는 페이로드만을 포함한 복제 패킷의 전송을 인에이블한다. 이는 테스트 목적으로만 사용된다.

TPP 하드웨어 실현

다음의 도 5는 PID 인식에 기초한 하드웨어 실현을 설명한다. 트랜스포트 패킷 헤더로부터의 PID는 관련 메모리에 배치된 한 세트의 PIDs(32)와 동시에 비교된다. 헤더 내의 PID가 관련 메모리에 저장된 PID에 대응할 경우, CAM 인코더는 단일 시스템에서 매치 신호(유효-PID) 및 CAM 어드레스(cam-ad)를 출력한다. 유효-PID는 트랜스포트 스트림 패킷 분석을 초기화하는데 사용되며, 이는 트랜스포트 스트림 헤더로부터 나온 다른 비트들을 로딩하기 위한 처리를 인에이블한다: AF 로드(트랜스포트-에러, 페이로드-유닛-시작-지시자, 적응 필드 플래그, 연속성 카운터 인덱스 및 해독화 모듈의 제어를 위해 보류된 다른 유형의 암호화 플래그). 유효_PID가 검출되자마자, cam-ad는 관련 속성을 찾는데 사용된다. 시퀀서(sequencer)는 SRAM 어드레스 발생기를 제어하는 상태 머신(state machine)을 시동시키기 위해 판독-SRAM-속성을 발생시킨다. 상태 머신은 cam-ad로부터 SRAM 어드레스를 생성하며, 한번은 하드웨어 속성을 검색하기 위해, 두 번째는 소프트웨어 속성을 판독하기 위해 SRAM을 2회 판독한다(하드웨어 변경 속성 및 소프트웨어 변경 속성을 분리함으로써 복잡한 보호 메카니즘의 사용을 피할 수 있음). 이 두가지로부터, 속성 및 비트 스트림 정보(헤더에서 나온 비트), EOP 인터럽트를 발생시키기 위한 조건이 평가된다. 그 후, 하드웨어 속성은 다음 패킷(동일 PID)의 수신용으로 갱신된다. 소프트웨어 속성은(스트림-id-필터와 같은) 다소 매우 정확한 기능을 위해서만 EOP 인터럽트 서비스 루틴의 일부로서 갱신될 수 있다. 속성이 다음 패킷의 시작 이전에 변경될 필요가 있을 경우, 비트율, 즉, 패킷 및 인터럽트 루틴 서비스 레이턴시 간의 갭은 이를 허용하기에는 몇몇 제약점을 고려하여야 한다.

패킷 분석기 제어 비트(도 4)에 의존하여, 패킷은 폐기되고, 최종 목적지까지 자동적으로 전송되거나 내장형 소프트웨어에 의해 사전 처리될 것이다. EOP 인터럽트를 발생시키는 이들 조건 중의 하나가 만족될 때, 인터럽트를 제어하는 비트는 활성 상태가 되고, 다음 패킷의 제 1바이트에 대해 리셋된다. EOP 인터럽트는 패킷의 앤드에 관한 인터럽트의 위치를 제어할 수 있다. EOP 인터럽트를 발생시키는 상이한 조건들(도 5: 논리 블록)에 관하여 설명한다.

1-에러 패킷

```

If (Te =1)
  if (En_erroneous_p = 1)
    EOP it
  else
    "discard packet"
end if

```

-완성 패킷을 삭제하기 위한 것보다 작은 에러를 인에이블하기 위해 시스템이 덜 파괴될 수 있는 경우에만 에러를 포함한 정상 패킷이 폐기된다. 그 경우, 패킷은 버퍼에 전송되고, 펌웨어는 패킷 형태(비디오, 오디오 등...)에 따라 최종 결정을 얻게 된다.

2-연속성 검출

```

elsif (Current_CC = Previous_CC)
  “duplicate packet”
  if ((AF = “01”) and (En_duplicate_p = 0))
    “discard packet”
  else
    EOP it
  end if
elsif (Current_CC /= Previous_CC +
1)
  EOP it

```

-EOP 패킷 인터럽트는 불연속성 카운터 검출 후에 발생될 것이다. MPEG2에서, CC 불연속성은 적응 필드 내의 불연속성 플래그가 세트될 때 가능하다. 이하 설명된 바와 같이, 적응 필드를 갖는 모든 패킷은 EOP 인터럽트를 발생시킨다. 페이로드(AF=1)만을 포함한 복제 패킷은 정상적으로 폐기된다. 그들은 내장형 소프트웨어로 인하여 글로벌 컨트롤 비트 En_복제_p가 세트되는지의 최종 결정을 얻을 수 있도록 전송될 수 있다.

3-다른 조건 리스트

EOP 인터럽트는

```

elsif (AF = “1x”)
  EOP it
:
elsif (Payload_unit_start_indicator =
1)
  EOP it
:
elsif ( Hpei = 1) and (En_Hpei = 1)
  EOP it
:
|
elsif ( Spei = 1)
  EOP it
elsif (AF = “01” and (((Hpei = 0) and (En_Hpei = 1)) or (Spei = 0)) and (stream-id = 1))  과

```

-적응 필드를 포함한 모든 패킷,

-페이로드-유닛-시작-지시자가 세트된 모든 패킷,

-페이로드-유닛-시작-지시자 세트를 갖는 패킷을 따르는 모든 패킷. 이는 PES 헤더가 1 패킷 이상을 얻을 수 있기 때문에 중요하다. 이러한 특성은 인에이블 또는 디스에이블될 수 있으며, Spei는 충분히 조기에 설정될 수 있다(도 6참조).

```

  “discard packet”
else
  “packet automatically
  transmitted”
end if

```

-내장형 소프트웨어가 추가 분석을 원하는 모든 패킷.

-패킷이 내장형 소프트웨어에 의해 분석되게 하는 속성이 설정되지 않는 한, 스트림-id가 활성 상태에 있을 경우(이전의 경우), 페이로드만을 포함한 모든 패킷은 폐기된다.

주목 알고리즘 내에 순차적으로 평가된 이들 조건의 전부는 실제로 하드웨어에 의해 나란히 계산된다.

일단 EOP 인터럽트 조건이 만족되면, 내장형 RISC 프로세서는 분석 업무에 착수한다.

트랜스포트 스트림 소프트웨어 분석

마이크로프로세서는 트랜스포트 패킷 파서 하드웨어 모듈보다 더욱 복잡한 분석을 행할 수 있으며, 하드웨어 사전 처리를 완성할 것이다. 부가적으로, 처리는 패킷에 대응하는 데이터 형태에 따라 다를 수 있으며, 사실상, 사용될 특정 알고리즘을 갖는 패킷에 관련하는 PID 값에 기초하여, 소프트웨어 테이블을 한정하는 것은 매우 쉽다.

트랜스포트 패킷의 소프트웨어 처리는 인터럽트에 의해 구동된다:

일단 EOP 인터럽트가 발생하면, 패킷을 분석하는 처리가 시작된다. LPU 레이턴시 응답 및 비트 스트리밍에 따라 상이한 두가지 처리 방법이 선택될 수 있다.

제 1의 경우, 소프트웨어가 PES 헤더를 포함한 패킷의 EOP 인터럽트를 수신하자 마자 TPP 하드웨어 모듈에서 Spei 플래그를 설정한다. 패킷의 처리 후, 소프트웨어는 PES 헤더가 다음의 트랜스포트 패킷에서 계속되는지의 여부를 검출할 수 있다: PES 헤더가 현재 패킷에서 끝날 경우, CPU는 다음의 패킷을 분석할 필요가 없기 때문에 패킷 분석기 제어 레지스터에서 Spei 비트를 리셋하고; 그렇지 않으면, Spei 비트는 변하지 않은 채로 남아 있다. 이 방법은 입력 비트율에 비해, CPU 레이턴시가 다음 패킷의 도착 이전에 Spei를 세트할 수 있을 경우에만 사용될 수 있다.

반대로, 제 2방법은 임의의 경우에 사용될 수 있으며: En_Hpei 비트가 패킷 분석기 컨트롤 레지스터 내로 세트될 경우, TPP 하드웨어가 PES 헤더 연속성을 포함한 패킷을 미리 ARM 프로세서에 송신하기 때문에 CPU는 각 PES 헤더 다음의 Spei 비트를 세트할 필요가 없다.

인터럽트의 원인이 TPP 하드웨어 모듈의 레지스터에서 플래그를 판독함으로써 발견된 후, 그 경우, 적응 필드가 처리된다.

적응 필드 분석 후, 패킷은 어떤 페이로드도 존재하지 않을 경우 폐기될 수 있어 페이로드는 이전 패킷으로부터 복제된다.

이점에서, 소프트웨어 처리의 실행은 패킷 형태(오디오, 비디오, 프라이비트 데이터, 서비스 정보, ...)에 따라 개별 루틴으로 분기된다. 그 후, 데이터는 DMA 채널을 통해 특정 버퍼로 전달된다.

이는 패킷(도 6참조)을 분석하는데 사용될 수 있는 알고리즘이다: 다음 리스트(Pes/Psi 시작 플래그의 검사 및 Spei 플래그의 세팅)의 제 1포인트가 선택된다.

사실상, 전술한 바와 같이, 소프트웨어 레이턴시가 TPP 하드웨어 모듈 분석전에 다음 트랜스포트 패킷을 세트할 수 있기에 충분히 짧을 경우, 이 동작은 CPU 간섭(intervention)을 최소화할 것이다: PES 헤더 시작을 따르는 EOP 인터럽트가 정말로 처리되어야 할 경우에만 EOP 인터럽트는 마이크로프로세서에 송신될 것이다.

달리 말하면, 인입 스트리밍의 비트율 및 CPU 레이턴시가 매우 높을 경우, Hpei 메카니즘이 인에이블(En_Hpei = 1)될 수 있고, PES 헤더 시작점을 따른 동일 PID를 갖는 다음 패킷은 항상 CPU로 전송될 것이다.

PES 헤더가 패킷 분석기 컨트롤 레지스터 내의 Pes/Psi 시작 플래그를 판독함으로써 현재 트랜스포트 패킷에서 시작하는지를 검사해 보자: 패킷 상태 워드(도 4참조).

PES 헤더가 현재 트랜스포트 패킷에서 시작할 경우, 패킷 분석기 컨트롤 레지스터 내의 Spei 비트는 동일 PID를 갖는 다음 패킷을 저지(intercept)하기 위해 CPU에 의해 세트되어야 한다: 사실상, PES 헤더는 다음의 트랜스포트 패킷에서 연속적일 수 있다.

패킷 분석기 컨트롤 레지스터 내의 AF 플래그를 판독함으로써 적응 필드의 존재를 검사해 보자: 패킷 상태 워드(도 4참조).

적응 필드가 패킷으로 전송될 경우, 관련 데이터가 처리되어야 한다. 특히, PCR은 기준 클록의 복구를 제어하기 위해 추출 및 필터링되어야 한다. 다른 중요한 부분은 불연속성 상태의 분석으로 구성된다: 적응 필드 내의 불연속성 상태 비트가 설정될 경우, 연속성 카운터 값 내의 불연속성 미싱(missing) 패킷에 대응하지 않는다. 불연속성 상태가 설정될 경우, 로컬 클록은 (존재한다면) PCR 값으로 재-초기화되어야 한다.

패킷이 패킷 분석기 컨트롤 레지스터 내의 복제 패킷 플래그를 판독함으로써 복제되는지를 검사해 보자(도 4참조).

패킷이 복제될 경우, 다른 처리가 요구되지 않으므로, 폐기된다. PES 헤더가 존재하고 Hpei 법이 사용되지 않기 때문에 Spei 비트가 패킷의 시작부에서 변경될 경우, Spei 비트는 처리의 종료 이전에 클리어되어야 한다.

패킷 분석기 컨트롤 레지스터 내의 AF 플래그를 판독함으로써 패킷이 트랜스포트 페이로드를 포함하는지를 검사해 보자(도 4참조).

페이로드가 없을 경우(적응 필드만), 다른 처리가 요구되지 않으므로 패킷은 폐기된다.

패킷 형태 검사

알고리즘의 이 부분은 특정 처리가 실행되도록 각각 선택된 PID에 관련된 테이블에 기초한다: 이러한 방식으로, 패킷은 관련 루틴에 의해 처리된다. 테이블은 32 엔트리, 각각의 가능한 선택 PID 중의 하나를 포함하며: 각 엔트리는 패킷 형태(즉, 오디오 패킷, 비디오 패킷, PES 프라이비트 데이터, PSI 등) 및 데이터가 저장되어야 하는 메모리 어드레스를 식별할 수 있게 한다. 특히, 패킷 형태가 비디오 스트리밍 또는 오디오 스트리밍에(일반적으로, 자동 DMA 전송에) 대응하고 Hpei 메카니즘이 사용되지 않을 경우, 패킷 처리 후에 Spei 비트는 PES 헤더가 종료되지 않을 경우에 클리어되어야 한다.

DMA 전송 시작

처리 후의 데이터는 특정 메모리 버퍼에 최종적으로 전송된다.

완성 알고리즘은 도 6에 도시되어 있다: 블랙 모듈은 Hpei 메카니즘이 디스에이블되는 경우에만 실행되는 알고리즘의 논리 블록에 대응한다.

예: PES 선택을 분석하지 않는 비디오 디코더에 대한 트랜스포트 스트리밍.

이 경우, 비디오 구성 요소 스트리밍만이 비디오 디코더로 송신된다: PTS 값이 추추출되어 비디오 디코더의

레지스터 내로 기록된다. DVB 추천에 의해 허용되므로, PES 헤더에 포함될 수 있는 모든 다른 필드는 고려되지 않는다. 도 7은 PES 패킷의 구성을 설명한다.

다음 알고리즘은 PES 패킷을 처리하는데 사용된다. PES 헤더의 제 1부분은 남은 필드를 처리하기 위해 임시 버퍼에 저장된다: 헤더가 상이한 패킷들 사이에서 분리될 경우, 제 2패킷이 수신된 후에 처리가 계속되고, 헤더의 제 1의 바이트들이 임시 버퍼 내로 기록되어야 한다.

이러한 예는 PES 분석에 관련된 부분만을 설명한다. CPU 처리의 다른 부분에 대하여, 상술한 알고리즘이 적용된다.

TPP 하드웨어 모듈의 PID 선택을 인에이블하기 전에 TPP 컨트롤 레지스터의 플래그는 다음과 같이 초기화 된다:

En_erroneous_p = 0

En_duplicate_p = 0

Spei = 0

stream-id filter flag = 0

PSI flag = 0

En_Hpei = 1

-에러를 갖는 패킷은 폐기된다.

-적응 필드를 운반하지 않는 복제 패킷은 폐기된다: 이 경우, 페이로드는 이전 패킷의 페이로드와 동일하다.

-CPU 간섭없이 자동 전송: 이전 선택에서 설명된 조건들이 발생될 때에만 CPU에 의해 패킷이 저지된다.

-스트림-id 필드 상의 어떤 필터도 요구되지 않는다.

-비디오 스트림은 PSI 섹션에 의해 전송되는 것이 아니라, PES 신트екс를 사용함으로써 전송된다.

-Hpei 메카니즘이 인에이블된다.

알고리즘은 다음의 의사-코드(pseudo-code)에 의해 설명될 수 있다.

PesStatus는 이전 상태의 PES 헤더 처리의 상태를 나타낸다(0 = PES 헤더가 종료되고; 1 = PES 헤더는 이전 패킷으로부터 계속되며, 아직 분석되지 않고 있고; 2 - 이전 패킷에서부터 연속되며, 모든 유용한 필드는 이미 분석되어 있다).

이전 사이즈는 이전 패킷에서 분석된 PES 헤더의 길이, 및 갱신 후, 이전 패킷에서 분석된 PES 패킷의 사이즈 + 현재 트랜스포트 패킷에 포함된 PES 패킷의 사이즈를 나타낸다.

```

if(CC-error flag = 1) and
  (discontinuity status = TRUE)
    "start error concealment"

```

```

if(PES/PSI start flag = 0) and
  (PesStatus = 0)
    "arrange starting point of the data
      to be transferred"

```

```

  elseif(PesStatus = 2)           /

```

-연속성 카운터에 에러가 있지만, 불연속성이 허용되지 않는다: 이전 패킷의 일부는 미스된다. 이 경우, 에러 잠복의 적정 방법이 시작되어야 한다.

- 이 패킷 내에 어떠한 PES 헤더도 없다: 모든 페이로드가 전송되어야 한다.

PayloadLength = 184 - AFlength

PreviousSize = PreviousSize + PayloadLength

```

if(PreviousSize < PES packet length)
  "no operation"

```

-PES 헤더 연속성이 있으나, 모든 유용한 필드는 이전 패킷에서 분석되어 있다.

-페이로드 길이를 바이트로 계산: 페이로드 사이즈는 트랜스포트 패킷 사이즈(184 바이트)-적용 필드 길이이다.

-사이즈 갱신.

-PES 헤더는 이 패킷에서 끝나지 않는다: PES 페이로드가 아직 시작하지 않았기 때문에, 어떤 데이터도 전송되지 않는다.

-PES 헤더는 이 패킷에서 끝난다: PES 페이로드가 전송되어야 한다.

```

else
  PesStatus = 0
    "arrange starting point of the data to be transferred"

```

else

-PesStatus는 2와 같다:PES 헤더는 아직 분석되지 않고 있다.

```
PayloadLength = 184 - AFlength
```

```
"append N bytes to temporary buffer" (N = 14 - PreviousSize)
```

```
PreviousSize = PreviousSize + PayloadLength
```

```
if (PreviousSize < 9)
```

```
    PesStatus = 1;
```

-페이지 길이를 바이트로 계산: 페이지 사이즈는 트랜스 포트 패킷 사이즈(184 바이트)

-적응 필드 길이이다.

-사이즈 갱신.

-PES 헤더의 고정된 부분은 이 패킷에 전부 포함되지는 않는다.

```
else
```

```
if (pts_flag = 0)
```

```
if (PreviousSize < PES header length)
    PesStatus = 2; No more useful fields but the PES header
    does not end in this packet.
```

```
else
```

```
    PesStatus = 0;
    "arrange starting point of the data to be transferred"
```

-이전 사이즈는 9 이상이다: PES 헤더의 고정된 부분의 모든 플래그는 이 패킷에 있다.

어떤 PTS도 PES 헤더에 의해 운반되지 않는다.

-PES 헤더는 이 패킷에서 끝난다: PES 페이로드는 전송되어야 한다.

```
else
```

```
if (PreviousSize < 14)
    PesStatus = 1;
```

```
else
```

```
    "write PTS to video decoder"
```

```
if (PreviousSize < PES header length)
```

```
    PesStatus = 2;
```

-PES 헤더는 PTS를 포함한다.

-PTS는 다음 패킷에 있다.

-PTS는 이 패킷에 있다.

-헤더는 이 패킷에서 끝나지 않는다.

else

PesStatus = 0;

“arrange starting point of the data to be transferred”

-헤더는 종료된다.

-PES 헤더는 이 패킷에서 끝난다: PES 페이로드는 전송되어야 한다.

이 경우, 패킷은 PES 헤더가 존재할 경우에만 CPU로 송신된다: 그러한 일은 트랜스포트 패킷 헤더 내의 PUS 지시자의 값을 검사함으로써 쉽게 발견될 수 있다.

비디오 스트림이 15Mbps의 비트율을 갖는 프레임으로 코드화된다고 가정해보자: CPU가 각 패킷을 처리해야 할 경우, 약 100 μ sec마다 인터럽트될 것이다.

반대로, 예비 필터와, TPP 하드웨어에 의해 실행된 루팅을 이용함으로써, CPU는 각 PES 패킷에서 인터럽트된다: PES 패킷 길이가 적어도 프레임 사이즈와 동일할 경우, 인터럽트는 최악의 경우, 40ms로 발생된다.

예: PES 신트랙을 분석하는 오디오 디코더에 대한 트랜스포트 스트림, 스트림-id 필드 상의 필터링이 행해진다.

이 예에서, 오디오 디코더는 PES 패킷을 분석할 수 있지만, CPU는 특정값과 다른 스트림-id를 가진 모든 PES 패킷을 필터링해야 한다: 스트림-id가 동일한 경우.

도 6에 도시되고 이미 설명한 일반적인 알고리즘이 사용된다: 다음 부분은 오디오 PES 패킷에 관한 처리의 특정 부분만을 설명한다. 특히, Hpei 메카니즘이 디스에이블되어, CPU는 PES 헤더의 시작점을 갖는 트랜스포트 패킷이 수신될 때마다 Spei 플래그를 세트하게 된다.

TPP 하드웨어 모듈의 PID 선택의 인에이블링하기 전에, TPP 컨트롤 레지스터는 다음과 같이 초기화된다:

En_errorous_p = 0

En_duplicate_p = 0

Spei = 0

stream-id filter flag = 1

PSI flag = 0

En_Hpei = 0

-에러를 갖는 패킷이 폐기된다.

-적응 필드를 운반하지 않는 복제 패킷이 폐기된다: 이 경우, 페이로드는 이전 패킷의 페이로드와 동일하다.

-CPU 간섭없는 자동 전송: 패킷은 이전 섹션에 설명된 조건이 발생될 때에만 CPU에 의해 저지된다.

-스트림-id-필드에 대한 필터가 요구된다.

-오디오 스트림은 PSI 섹션에 의해 전송되는 것이 아니라, -PES 신트랙의 사용으로 전송된다.

-Hpei 메카니즘이 디스에이블되고, CPU는 세트 및

Spei 비트를 리셋해야 한다.

초기화 후, TPP 하드웨어 모듈은 제 1PES 헤더가 수신될 때까지 모든 패킷을 폐기한다. 이 점에서, EOP 인터럽트가 발생되고, CPU는 패킷을 분석한다.

스트림-id 값이 특정 패턴에 일치할 경우, 모든 PES 패킷은 오디오 디코더로 송신되어야 한다: 결과적으로, 마이크로프로세서는 패킷 분석기 컨트롤 레지스터에서 스트림-id 필터 플래그를 클리어하고, 모든 트랜스포트 패킷은 자동적으로 오디오 디코더에 전송된다. 다른 PES 헤더가 TPP 하드웨어에 의해 발견될

(트랜스포트 헤더 내의 패킷 유닛 시작 지시자가 세트될) 때. 인터럽트가 발생되고, CPU는 새로운 PES 패킷의 스트림-id 값에 대한 다른 필터 동작을 행할 수 있다.

스트림-id 값이 특정 패턴에 매치되지 않을 경우, CPU는 TPP 모듈에서 스트림-id 플래그를 세트하고, 동일 PID를 갖는 모든 다음의 트랜스포트 패킷은 다음 PES 헤더까지 하드웨어에 의해 자동적으로 폐기된다.

알고리즘은 다음 의사-코드에 의해 설명될 수 있다. CPU는 모든 PES 헤더를 분석하지 않고 스트림-id 필드만을 분석한다.

PesStatus는 이전 상태의 PES 헤더 처리의 상태를 나타낸다(0=PES 헤더가 종료되거나 PES 헤더는 이전 패킷에서부터 계속되지만, 스트림-id 값은 이미 필터링되어 있고; 1=PES 헤더는 이전 패킷에서부터 계속되며 스트림-id 값은 아직 필터링되어 있지 않다).

이전 사이즈는 이전 패킷에서 판독된 PES 헤더의 길이를 나타낸다.

-현재 패킷은 스트림-id 필터 패턴에 일치하지 않으며 어떤 PES 헤더도 없다: 패킷은 폐기되어야 한다.

-시작 에러 잠복

```

if(stream-id filter flag = 0) and
  (PES/PSI start flag = 0)
    "discard the packet"

else
  if(CC-error flag = 1) and
    (discontinuity status      =
      TRUE)

```

```

if(PES/PSI start flag = 0) and
  (PesStatus = 0)
    "arrange starting point of
    the          data      to      be
    transferred"

```

else

PayloadLength=184-AFlength

PreviousSize = PreviousSize + PayloadLength

if(PreviousSize < 4)

PesStatus = 1;

-연속성 카운터 에러가 있지만 불연속성이 허용되지 않는다: 이전 패킷의 일부는 미스된다. 이 경우, 에러 잠복의 적정 방법이 시작되어야 한다.

-이 패킷에 어떤 PES 헤더도 없다: 이 모든 페이로드가 전송되어야 한다.

-PES 헤더 또는 PES 헤더 연속성이 있다.

-페이로드 길이를 바이트로 계산: 페이로드 사이즈는 트랜스포트 패킷 사이즈(184 바이트)-적응 필드 길이이다.

-사이즈 갱신.

-스트림-id는 이 패킷에 포함되지 않는다.

-다음 트랜스포트 패킷이 분석되어야 한다.

else

PesStatus = 0;

if (stream-id = “pattern”)

-이전 사이즈는 40이상이다: 스트림

-id 필드는 이 패킷에 포함되지 않는다.

-CPU는 다음 패킷을 분석할 필요없다.

-스트림-id 필드를 필터링: 그 값이 특정 패턴과 같을 경우, PES 패킷이 선택된다.

“clear stream-id filter flag”

“arrange starting point of the data to be transferred”

-TPP 하드웨어에 의해 패킷의 자동 폐기를 디스에이블.

-PES 헤더는 이 패킷에서 끝난다: PES 페이로드가 전송되어야 한다.

else

“set stream-id filter flag”

“discard the packet”

-스트림-id 필드는 특정 패턴과 다르다: PES 패킷이 폐기되어야 한다.

-TPP 하드웨어에 의해 패킷의 자동 폐기를 인에이블.

이러한 점에서, Spei 비트는 PES 헤더의 고려된 부분이 이미 처리되었으므로 PesStatus가 0과 같을 경우에만 클리어된다.

이 경우, 인터럽스는 스트림-id 값을 필터링하기 위해 PES 헤더가 존재할 때에만 CPU로 송신된다. 스트림-id가 선택된 값과 일치할 경우, 다음 패킷들은 목적지 버퍼에 자동적으로 송신되며; 반대로, 스트림-id가 선택된 값과 일치하지 않을 경우, 모든 다음 패킷들은 다음 PES 헤더까지 TPP 하드웨어 모듈에 의해 폐기된다.

이러한 메카니즘이 없으면, CPU는 스트림-id의 불필요한 값을 갖는 PES 패킷에 포함된 모든 트랜스포트 패킷을 폐기하기 위해 인터럽트를 수신하도록 강요될 것이다.

발명의 효과

본 발명에 따르면, MPEG2 디코더에 사용되는 IMPEG2 트랜스포트 스트림 패킷 파서 시스템을 제공할 수 있으며, 이 시스템은 관련 플래그로의 사전 처리한 후의 패킷을 갖는 중간 버퍼, 및 송신 플래그에 의해 선택된 패킷의 추가 소프트웨어 처리를 위한 프로세서를 포함한다.

(57) 청구의 범위

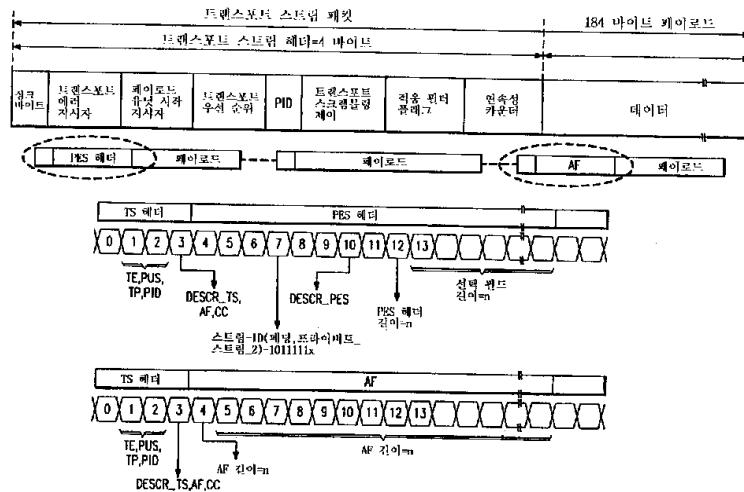
청구항 1

관련 플래그로 사전 처리한 후의 패킷을 갖는 중간 버퍼, 및

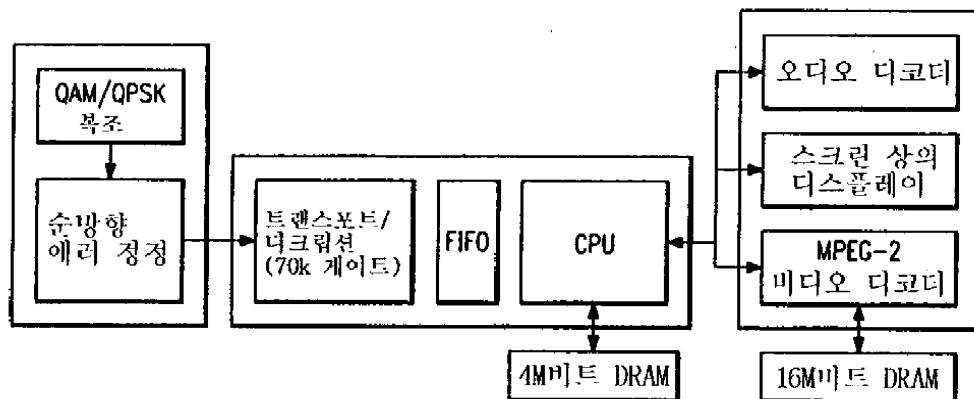
송신 플래그에 의해 선택된 패킷의 추가 소프트웨어 처리를 위한 프로세서를 포함하는 트랜스포트 스트림 파서 시스템(transport stream parser system).

도면

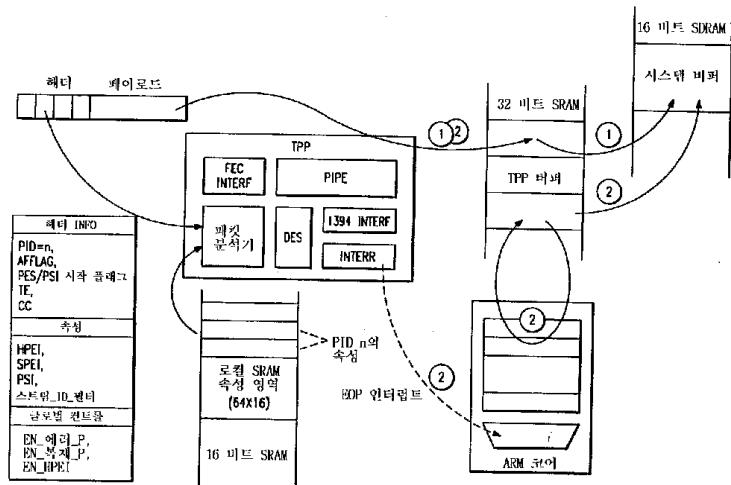
도면1



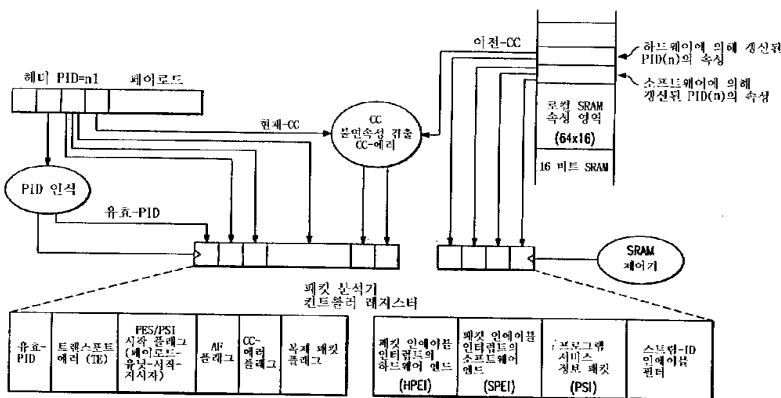
도면2



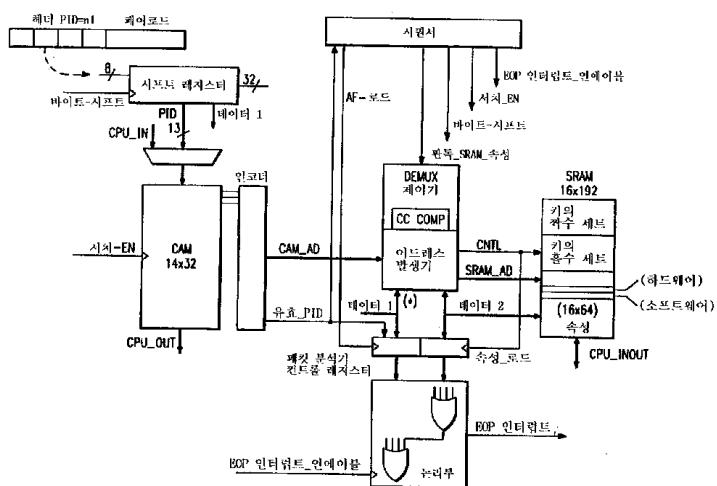
도면3



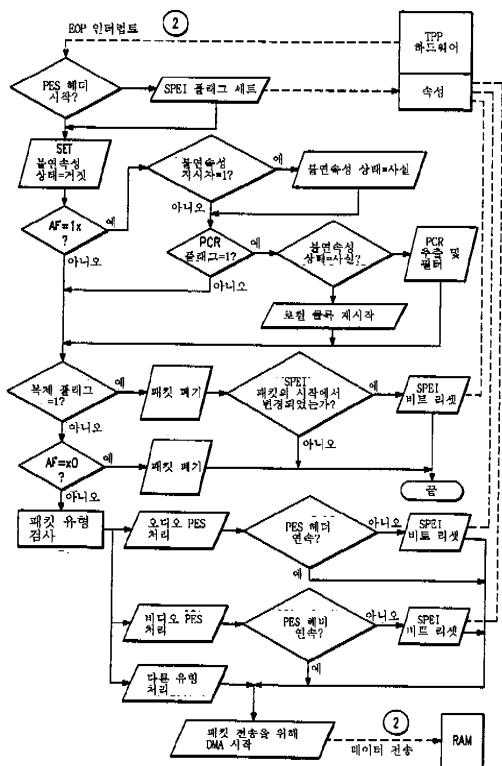
도면4



도면5



도면6



도면7

