Slide 0

7장 그밖의 관련된 주제들

Interrupt

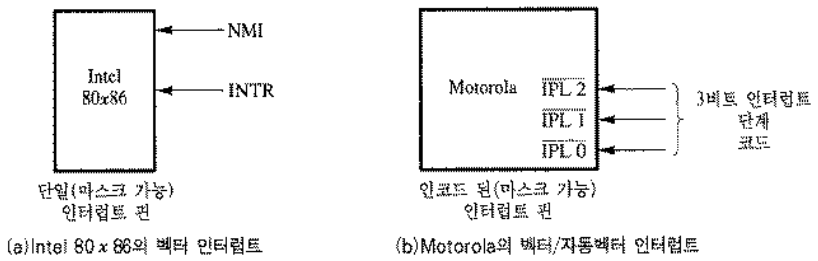
Slide 1

- 외부 인터럽트
  - 인터럽트의 종류
    - direct interrupt
      1. CPU 에 인터럽트 신호가 직접들어감
      2. CPU 내부의 인터럽트 레지스터 사용 벡터 결정
      3. ISR 수행
      4. 인터럽트 처리가 빠르다 (ack. 싸이클이 필요없음)
    - indirect interrupt
      1. CPU 에 인터럽트 요청
      2. CPU 가 인터럽트 acknowledge 할 때 해당벡터 제공
      3. 인터럽트 처리가 느리다 (ack. 싸이클 필요)
  - 인터럽트
    - 외부 장치에서 신호발생 (interrupt request)
    - CPU 인터럽트 인지 및 ACK (interrupt acknowledge)
    - 현재 수행중이던 프로그램 정보 저장
    - 인터럽트 서비스 루틴 처리
    - 수행중이던 프로그램으로 복귀

Interrupt (Cont'd)

Slide 2

- 마스크 불가능 인터럽트 (Nonmaskable Interrupt (NMI))
  - 소프트웨어로 마스크 불가능한 인터럽트
  - 대부분의 MPU에서 INTA 생성필요 없음 (autovectored interrupt)
  - 미리 지정된 메모리로 이동
  - 이미 진행중이던 인터럽트도 무시될수 있음
- 마스크 가능 혹은 사용자 인터럽트 (그림 7.3)



  - Intel 계열 : INTR (범용 인터럽트 입력핀)
  - Motorola 계열: IPL0# ~ IPL2# 의 인코드된 입력핀 사용 (000: NMI)

Interrupt (Cont'd)

Slide 3

- 벡터 인터럽트
  - 대부분의 프로세서가 사용
  - 인터럽트 요청 (IREQ)
  - MPU 가 갖추어야할 기능들
    1. 실행중이던 프로그램으로의 복귀 기능
    2. 여러 장치중 인터럽트 발생장치 확인 기능
    3. 여러장치에서 동시에 인터럽트 발생시 우선순위 정하는 기능
  - 인터럽트 종류
    1. 외부 I/O (하드웨어) 인터럽트
    2. 소프트웨어 인터럽트나 예외
    3. 내부 하드웨어 예외 ex) divide by zero
  - 우선순위 (그림 7.4)
    1. RESET, BERR : clock cycle 의 끝에서 처리
    2. 특권명령, 불법명령 : 버스 싸이클의 끝에서 처리
    3. 소프트웨어 인터럽트 : 해당 명령어 싸이클 내에서 처리
    4. 사용자 I/O 인터럽트 : 현재 명령어 사이클의 끝에서 처리

Interrupt (Cont'd)

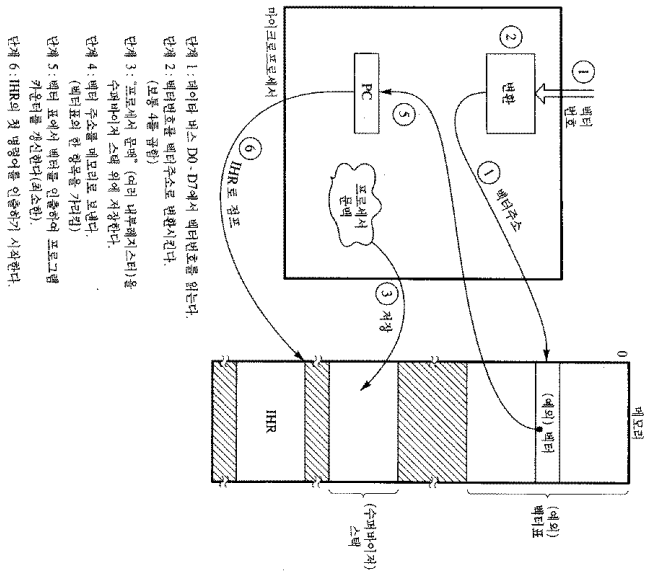
Slide 4

* 처리 순서
  1. 수퍼바이저 모드로 바꾼다
  2. 예외의 원인을 확인한다
  3. 현재의 프로세서 문맥을 저장
  4. 예외처리 루틴으로 분기
* 예외의 원인들
  1. 내부 하드웨어 예외 : 자동벡터 사용
  2. 소프트웨어 예외 : 자동벡터 사용
  3. I/O 하드웨어 인터럽트
     (a) 현재의 명령 완료후 인식
     (b) 현재의 프로세서 레벨보다 작은 것은 처리가 연기됨
     (c) 자동벡터인가 외부벡터 인터럽트인가를 구분
* 자동벡터 : 벡터 번호를 생성할수 없는 단순한 장치를 보조
* 벡터 인터럽트 : 벡터 번호를 생성할수 있는 장치들
  1. 인터럽트 인지싸이클에서 벡터번호를 제공

Interrupt (Cont'd)

Slide 5

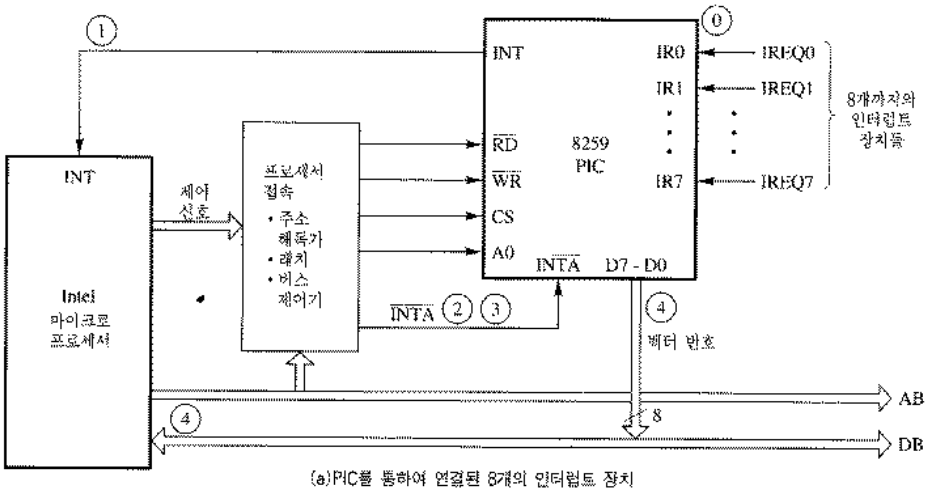
* 인터럽트 서비스 루틴으로 점프 단계 (그림 7.5)



Interrupt (Cont'd)

Slide 6

* 인텔의 벡터 인터럽트 구조
  1. 8259를 사용한 접속 (그림 7.6)



(a)PIC를 통하여 연결된 8개의 인터럽트 장치

Interrupt (Cont'd)

Slide 7

2. 64개의 I/O 인터럽트를 제공하기 위한 접속 (그림 7.7)

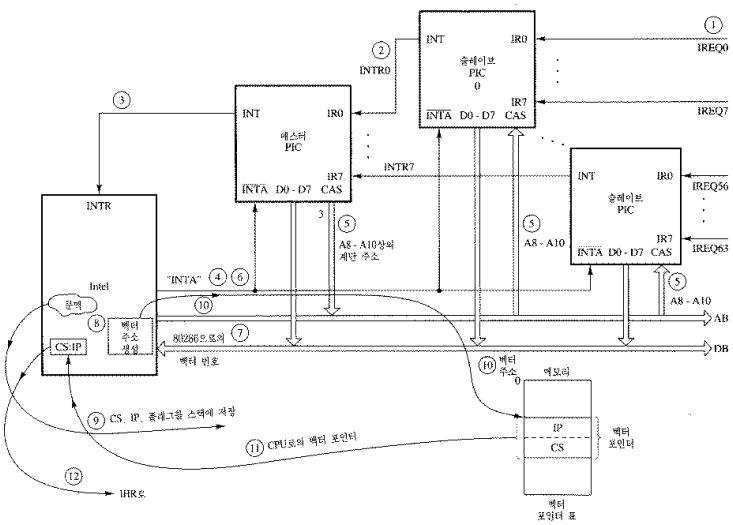


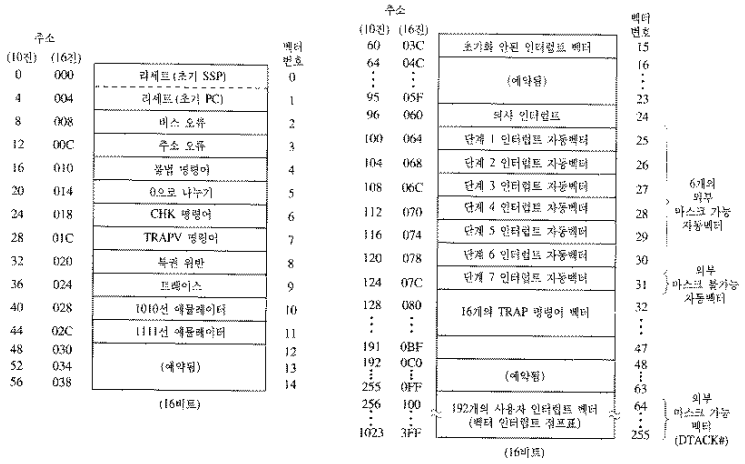
그림 7.7 제단 접속된 PIC로 연결된 Intel 마이크로프로세서 (동그라미친 번호가 매겨진 단계는 절차 7.2의 단계에 해당함).

Interrupt (Cont'd)

Slide 8

* 모토롤라의 벡터 인터럽트 구조

1. 68000의 벡터테이블 (그림 7.8)



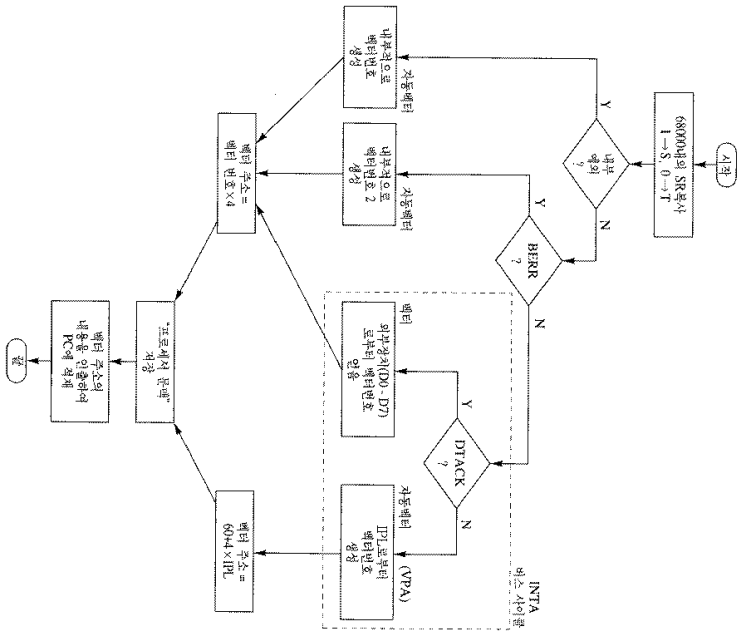
Interrupt (Cont'd)

Slide 9

2. IPL0# ~ IPL2#
   (a) 111 : 가장 낮은 우선순위 (인터럽트 없음)
   (b) 000 : 가장 높은 우선순위 (마스크 되지 않음)
3. 68000의 예외 처리 순서 (그림 7.9)

Interrupt (Cont'd)

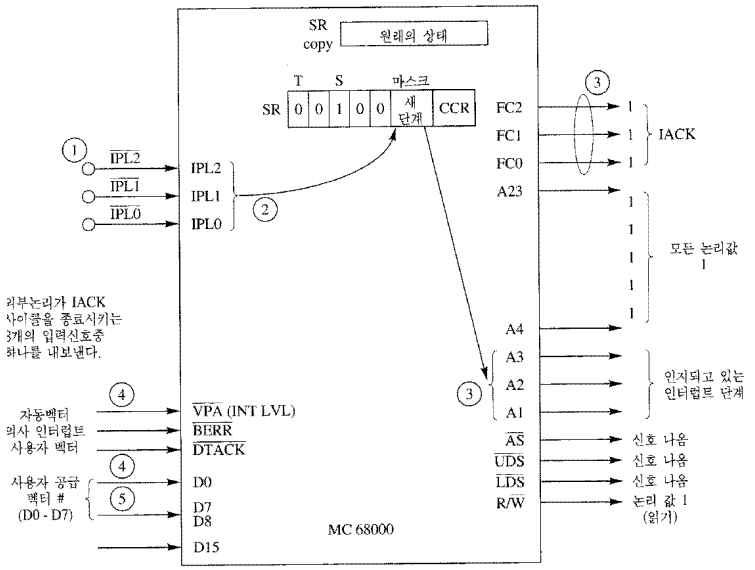
Slide 10



Interrupt (Cont'd)

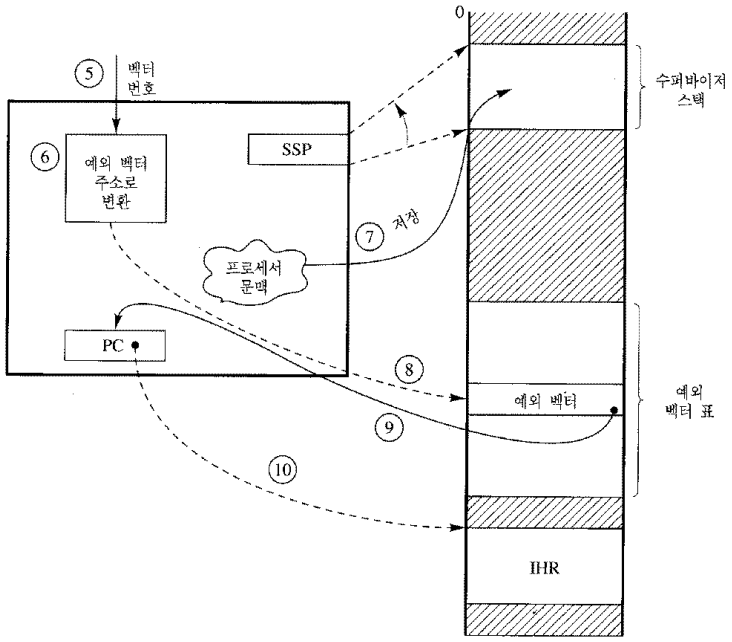
Slide 11

4. 모토롤라 계열의 예외 처리 단계 (그림 7.10)



Interrupt (Cont'd)

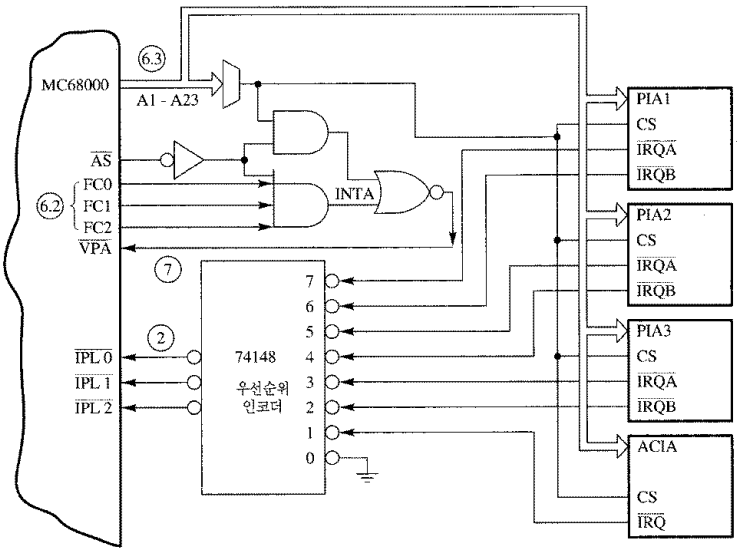
Slide 12



Interrupt (Cont'd)

Slide 13

5. 자동벡터 장치의 접속 (그림 7.11)



Interrupt (Cont'd)

Slide 14

6. 벡터장치와 자동벡터 장치의 접속 (그림 7.12)