Slide 0

11장. 입출력 구조

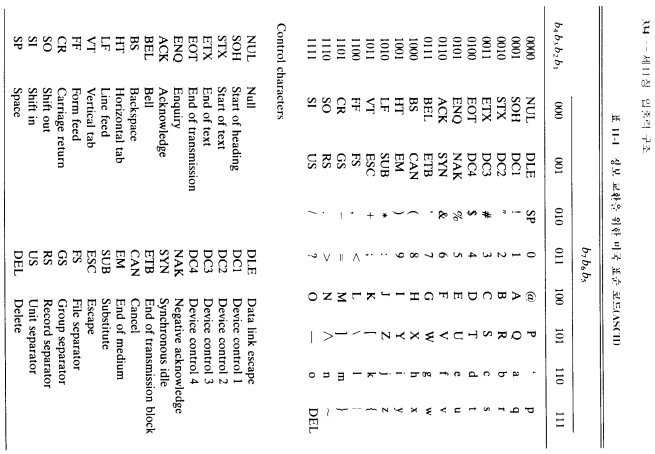
주변 장치

Slide 1

- 주변 장치의 기능
  - 외부와의 효율적인 통신 방법 제공
- 주변 장치의 종류
  - 입력 장치
    - 키보드, 마우스, 아날로그 디지탈 컨버터
  - 출력 장치
    - 디스플레이 장치 (모니터)
    - 프린터
- ASCII 영숫자 문자
  - 문자 숫자 기호 및 특수 기호의 이진 표준 코드 (표 11-1)

주변 장치 (계속)

Slide 2



34 — 제11장 입출력 구조

표 11-1 정보 교환을 위한 미국 표준 코드(ASCII)

$b_4b_3b_2b_1$	$b_7b_6b_5$ = 000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

Control characters

NUL	Null	DLE	Data link escape
SOH	Start of heading	DC1	Device control 1
STX	Start of text	DC2	Device control 2
ETX	End of text	DC3	Device control 3
EOT	End of transmission	DC4	Device control 4
ENQ	Enquiry	NAK	Negative acknowledge
ACK	Acknowledge	SYN	Synchronous idle
BEL	Bell	ETB	End of transmission block
BS	Backspace	CAN	Cancel
HT	Horizontal tab	EM	End of medium
LF	Line feed	SUB	Substitute
VT	Vertical tab	ESC	Escape
FF	Form feed	FS	File separator
CR	Carriage return	GS	Group separator
SO	Shift out	RS	Record separator
SI	Shift in	US	Unit separator
SP	Space	DEL	Delete

입출력 인터페이스

Slide 3

- 내부 저장장치와 외부 입출력 장치간의 이진정보 전송 방법 제공
- 컴퓨터와 주변 장치와의 가능한 차이점 →인터페이스 모듈이 해결
  1. 외부 장치의 신호값을 전기신호로 변환이 필요
  2. 전송속도의 차이, 동기화 절차가 필요
  3. 데이타 코드와 형식의 차이
  4. 주변장치의 동작별로 제어
- I/O 버스와 인터페이스 모듈
  - I/O 버스와 입출력 장치의 연결 (그림 11-1)

입출력 인터페이스 (계속)

Slide 4

어떤 특정한 장치와 통신하기 위해 CPU는 버스에 장치 주소를 지정하고, 이

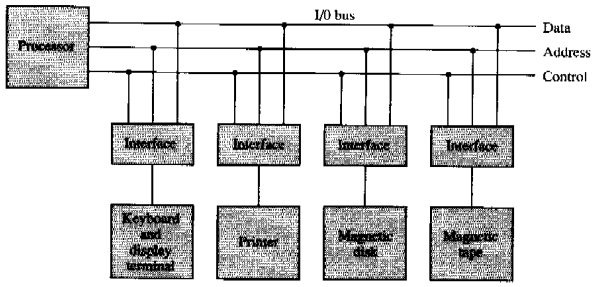


그림 11-1 I/O버스와 입출력 장치의 인터

- 인터페이스
  - CPU 부터 받은 명령을 해석하고 주변 장치 제어기에 신호를 보냄
  - 데이타의 흐름을 동기화 하고 전달속도를 관리
- 각 장치는 자기자신의 전기 기계적 장치를 제어하는 제어기를 갖음
- 3개의 버스를 이용 주변장치와 통신

입출력 인터페이스 (계속)

Slide 5

  - 주소를 내 보내면 주소에 해당하는 장치만 활성화
  - 그외 모든 다른 장치는 비활성화 됨
  - 활성화된 장치의 제어기와의 통신을 통하여 동작
- 장치 제어기의 명령 종류
  - 제어 명령 : 수행할 명령어
  - 테스트 명령 : 주변 장치의 상태 테스트 명령어
  - 데이타 출력 명령 : 데이타를 장치가 받아들이라는 명령
  - 데이타 입력 명령 : 데이타를 장치가 보내라는 명령

- I/O 대 메모리 버스
  - 프로세서와 메모리사이의 통신
  - 메모리및 I/O 와 통신하는 3가지 방법
    1. 메모리와 I/O에 대해 분리된 버스 사용
    2. 메모리와 I/O에 대해 공통의 버스 사용, 분리된 제어라인 사용
    3. 메모리와 I/O에 대해 공통의 버스와 공통의 제어라인 사용
- 격리형 대 메모리 맵형 I/O

입출력 인터페이스 (계속)

Slide 6

- 격리형 I/O 방식
  - * 공통의 버스 사용, 메모리 와 I/O 전송 구별은 제어라인을 통함
    →격리형 I/O 방식 (isolated I/O method) 라고 부름
  - * 메모리 명령어와 구별된 I/O 명령어 사용
- 메모리 맵형 I/O 방식
  - * 메모리와 동일 명령어 및 동일 제어라인을 사용
  - * 메모리와 입출력 사이의 구별이 없음

• I/O 인터페이스의 예 (그림 11-2)

입출력 인터페이스 (계속)

Slide 7

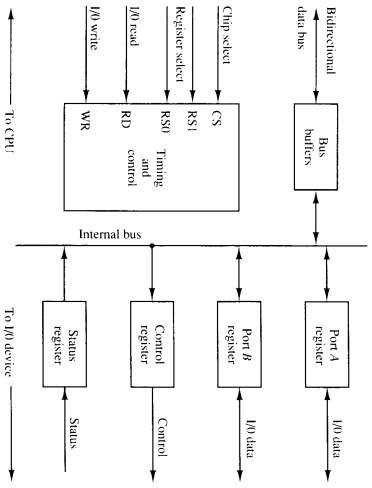


그림 11-2 I/O 인터페이스 예

CS	RS1	RS0	Register selected
0	×	×	None: data bus in high-impedance
1	0	0	Port A register
1	0	1	Port B register
1	1	0	Control register
1	1	1	Status register

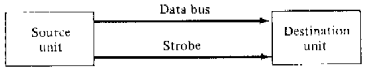
비동기 데이타 전송

Slide 8

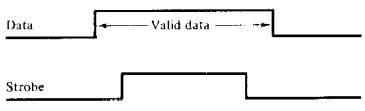
- 통신할 두개의 장치가 같은 클럭을 사용 →동기식 데이타 전송
- 통신할 두개의 장치가 다른 클럭을 사용 →비동기식 데이타 전송
- 비동기 데이타 전송
  - 데이타 전송 시각을 알리는 제어신호가 교환되어야함
  - 두가지 방식
    - 스트로브 펄스 (strobe pulse) 에 의한 방법
    - 핸드쉐이킹 (handshaking) 방법
- 스트로브 제어
  - 단 하나의 제어라인을 사용
  - strobe 신호 : 유효한 데이타가 있음을 알림
  - 메모리와 CPU 사이의 정보교환시 사용될수 있음
  - 소스개시의 전송 방식 (그림 11-3)

비동기 데이타 전송 (계속)

Slide 9



(a) Block diagram



(b) Timing diagram

그림 11-3 데이타 전송을 위한 소스 개시의 스트로브

- 목적지 개시의 전송 방식 (그림 11-4)

비동기 데이타 전송 (계속)

Slide 10

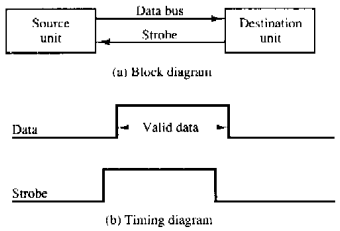


그림 11-4 데이터 전송을 위한 복식 제시의 소스 프로그램

하여 이루어진다.

- 핸드쉐이킹
  - 스트로브제어의 단점
    - 송신 장치는 수신장치가 데이타를 받아들였는지 알수 없음
    - 전송을 시작한 수신장치는 송신장치가 데이타를 버스에 놓았는지 알수없음
  - 해결 방법 →핸드쉐이킹 방법

비동기 데이타 전송 (계속)

Slide 11

- 소스개시의 데이타 전송 과정 (그림 11-5)

비동기 데이타 전송 (계속)

Slide 12

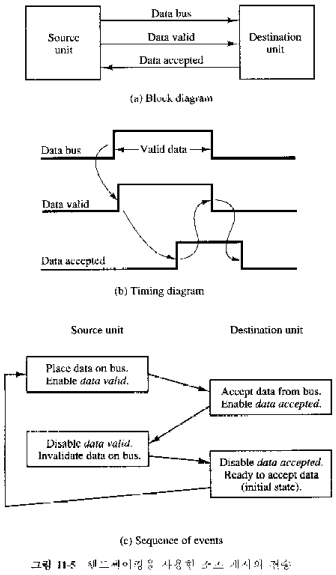


그림 11-5 핸드쉐이킹을 사용한 소스 개시의 전송

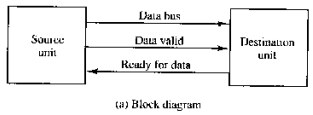
비동기 데이타 전송 (계속)

Slide 13

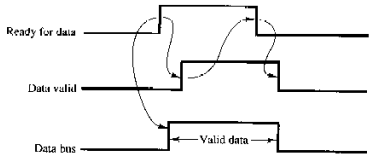
* 목적지 개시의 데이타 전송 과정 (그림 11-6)

비동기 데이타 전송 (계속)

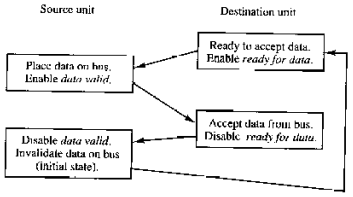
Slide 14



(a) Block diagram



(b) Timing diagram



(c) Sequence of events

그림 11-6 핸드-세이킹을 사용한 직렬적 제어의 전송

비동기 데이타 전송 (계속)

Slide 15

- 장점 →높은 융통성과 신뢰성을 갖음
- 전송시 한 장치가 이상이 있을경우 →타임아웃 방식으로 에러 검출

• 비동기 직렬 전송

- 병렬 전송
  * n 비트가 n 개의 선을 통해 동시에 전달
    →속도 빠름, 비용 큼
- 직렬 전송
  * 1 비트가 1개의 선을 통해 전달
    →속도 느림, 비용 작음
- 비동기 직렬 전송
  * 문자 코드의 양 끝에 특수한 비트가 들어감
  * 구성 (그림 11-7)

비동기 데이타 전송 (계속)

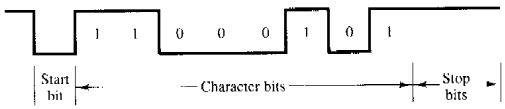


그림 11-7 비동기 직렬 전송

Slide 16

* 수신부 규칙
  1. 데이타가 없을때는 1-상태 유지
  2. 시작비트 (0-상태)로 데이타 시작알림
  3. 시작비트후에 데이타 정보 있음
  4. 마지막 데이타비트 후 최소한 1비트의 정지 비트
* 보레이트 (baud rate) →직렬 정보가 전송되는 속도
  →보통 초당 전송되는 데이타 비트
* 비동기 직렬 전송 회로 →universal asynchronous receiver transmitter (UART)

- 비동기 통신 인터페이스

비동기 데이타 전송 (계속)

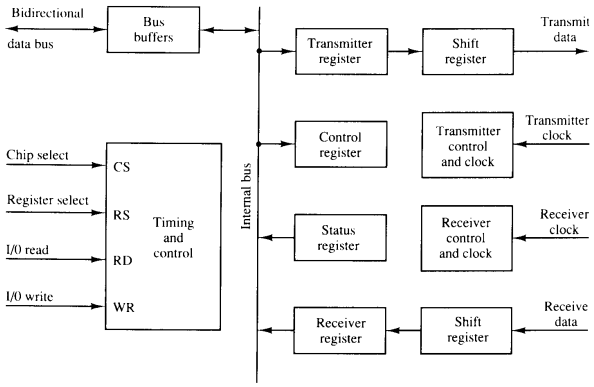
Slide 17

– 비동기 인터페이스 구성 (그림 11-8)

비동기 데이타 전송 (계속)

Slide 18

346 — 제 11 장 입출력 구조



CS	RS	Operation	Register selected
0	×	×	None: data bus in high-impedance
1	0	WR	Transmitter register
1	1	WR	Control register
1	0	RD	Receiver register
1	1	RD	Status register

그림 11-8 대표적인 비동기 통신 인터페이스의 블럭도

비동기 데이타 전송 (계속)

Slide 19

- 비동기 전송시 발생할수 있는 에러
  - 패리티 에러 (parity error) : 패리티가 안 맞을때
  - 프레밍 에러 (framing error) : 정지 비트의 갯수가 안 맞을때
  - 오버런 에러 (overrun error) : 수신레지스터를 읽지전에 다음 문자가 들어올때

- FIFO 버퍼
  - 첫번째 입력이 첫번째 출력이됨
  - 입력속도와 출력속도가 다를수 있음

    →비동기 전송에 유용하게 쓰임
  - 4 × 4 FIFO 버퍼 (그림 11-9)

비동기 데이타 전송 (계속)

Slide 20

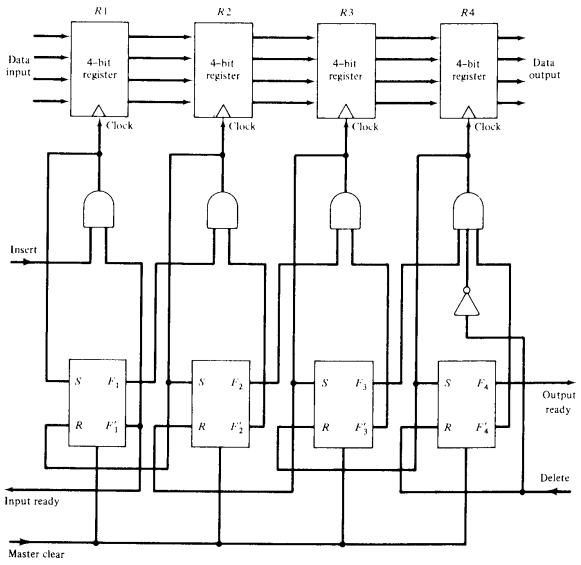


그림 11-9 4×4 FIFO 버퍼의 회로도

전송 모드

Slide 21

- 3가지 전송모드
  1. 프로그램된 I/O
  2. 인터럽트에의한 I/O
  3. 직접 메모리 접근 (DMA: Direct Memory Acess)
- 프로그램된 I/O
  - 프로그램명령어에 의하여 송수신
  - 계속 주변장치를 감시하여 송수신 가능한지를 보고 송수신
  - 송수신 가능한 상태가 아니면 가능한 상태를 점검하는 루프 수행
  - 예)
    * I/O 장치에서 CPU 로의 데이타 전송 (그림 11-10)

전송 모드 (계속)

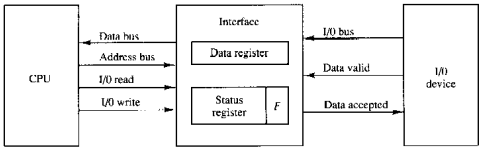


그림 11-10 I/O 장치에서 CPU 로의 데이타 전송

Slide 22

* 데이타를 입력받는 CPU 프로그램에 대한 흐름도

프로세서의 시간 낭비 →인터럽트 방식으로 해결

- 인터럽트에 의한 I/O
  - 프로그램 계속 수행
  - 주변장치가 송신가능하거나 수신한 데이타가 있을때 인터럽트
  - 인터럽트 서비스 루틴으로 가서 송수신 수행

전송 모드 (계속)

  - 벡터 인터럽트와 비벡터 인터럽트가 있음

Slide 23

- DMA에 의한 I/O
  - CPU 가 DMA 에게 데이타 송수신 요구
  - DMA 가 수행

우선순위 인터럽트

Slide 24

- 우선순위의 필요성
  - 여러개의 입출력 장치중에서 인터럽트 발생 장치 인지
  - 동시에 인터럽트가 들어왔을때 처리순서 우선순위 결정
- 데이지 체인 우선순위 인터럽트
  - 인터럽트 발생장치를 직렬로 연결
  - CPU 에 근접한 순서대로 우선순위 결정됨
  - 구성 (그림 11-12)

우선순위 인터럽트 (계속)

Slide 25

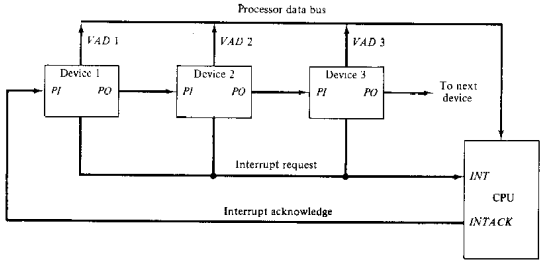


그림 11-12 데이지 체인 우선순위 인터럽트

  - 우선순위의 논리 회로 (그림 11-13)

우선순위 인터럽트 (계속)

Slide 26

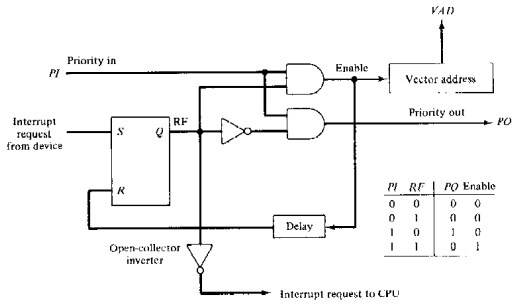


그림 11-13 한 단의 데이지 체인 우선순위 배선

버퍼를 요구 라인은 모든 장치에 공통이며 와이어드 논리(wired logic) 인결로

- 병렬 우선순위 인터럽트
  - 우선순위 인코더에 의하여 라인별로 우선순위가 결정됨
  - 우선순위 인터럽트 하드웨어 (그림 11-14)

우선순위 인터럽트 (계속)

Slide 27

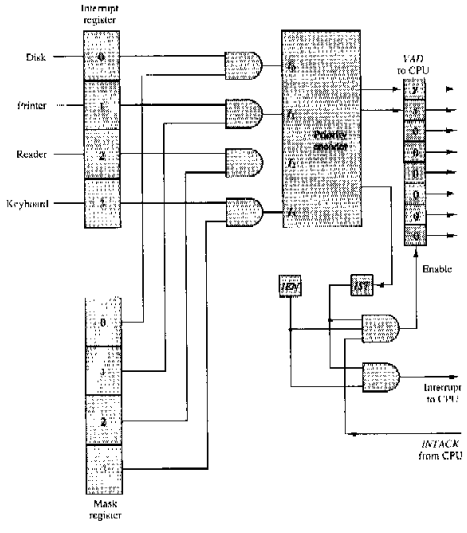


그림 11-14 우선순위 인터럽트 하드웨어

우선순위 인터럽트 (계속)

Slide 28

– 우선순위 인코더 진리표 (표 11-2)

표 11-2 우선순위 인코더 진리표

Inputs				Outputs			
I_0	I_1	I_2	I_3	x	y	IST	Boolean functions
1	×	×	×	0	0	1	
0	1	×	×	0	1	1	$x = I_0' I_1'$
0	0	1	×	1	0	1	$y = I_0' I_1 + I_0' I_2'$
0	0	0	1	1	1	1	$(IST) = I_0 \mid I_1 \mid I_2 \mid I_3$
0	0	0	0	×	×	0	

- 인터럽트 사이클

우선순위 인터럽트 (계속)

Slide 29

SP ←SP + 1	스택포인터의 증가
M[SP] ←PC	PC 를 스택에 push
INTACK ←1	인터럽트 승인 응답을 활성화
PC ←VAD	벡터 주소를 PC 에 전송
IEN ←0	더이상의 인터럽트를 금지
Go to fetch cycle	

- 소프트웨어 루틴
  - 인터럽트가 발생시 서비스할 프로그램 루틴 →인터럽트 서비스 루틴 (ISR)
  - 구성 예) (그림 11-15)

우선순위 인터럽트 (계속)

Slide 30

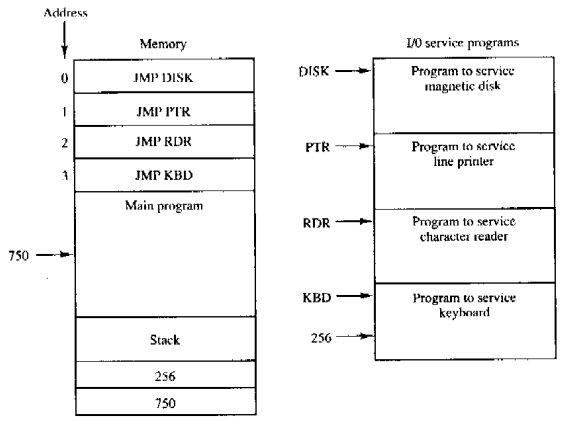


그림 11-15 인터럽트된 서비스 요청 레지스터에 저장된 프로그램

- 초기와 최종 동작
  - 인터럽트 서비스 루틴의 초기 동작

우선순위 인터럽트 (계속)

Slide 31

1. 보다 낮은 단계의 마스크 레지스터를 0 으로 함
2. 인터럽트 상태 비트 IST 를 클리어 함
3. 프로세서의 레지스터의 내용을 보관함
4. 인터럽트 인에이블 비트 IEN 을 세트
5. 서비스 루틴을 수행

- 인터럽트 서비스 루틴의 최종 동작
  1. 인터럽트 인에이블 IEN 을 클리어 함
  2. 프로세서 레지스터의 내용을 다시 저장
  3. 서비스된 자원에 해당하는 인터럽트 레지스터의 비트를 클리어
  4. 마스크 레지스터의 보다 낮은 우선순위의 비트를 세트
  5. 되돌아갈 주소를 PC 에 저장하고 IEN을 세트

직접 메모리 접근 (DMA)

- DMA의 필요성
  - 자기 디스크와 같은 고속의 저장 장치와 메모리 간의 고속의 데이타 전송시
  - CPU 가 직접 데이타 전송에 참여하면 속도가 느림, CPU 단순작업 수행
    →외부 데이타 전송 전용 장치 (DMA) 설치
  - DMA 가 버스를 구동시 →CPU 는 버스 동작 수행 불가

Slide 32

- DMA 전송을 위한 CPU 버스 신호 (그림 11-16)

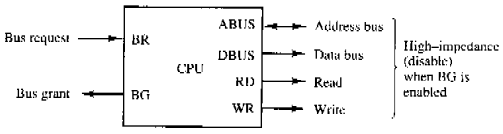


그림 11-16 DMA 전송을 위한 CPU 버스 신호

CPU는 다양한 방법으로 유지(idle) 상태에 있게 되는데 가장 일반적인 방법

  - DMA 가 BR (bus request) 로 버스 요구

직접 메모리 접근 (DMA) (계속)

  - BG 로 CPU 가 버스 사용 승인
  - CPU 는 모든 버스구동 선을 고 저항 상태 (high-impedance) 로 놓음
  - DMA 버스 사용 후 BR 를 inactive 로 놓음
  - CPU 버스 사용 가능

Slide 33

- 두가지 DMA 전송 모드
  - DMA 대량 전송 (burst transfer) →대량의 데이타를 전송
  - 사이클 스틸링 (cycle stealing) →한번에 한 데이타 워드를 전송
- DMA 제어기
  - DMA 제어기의 블럭도 (그림 11-17)

직접 메모리 접근 (DMA) (계속)

Slide 34

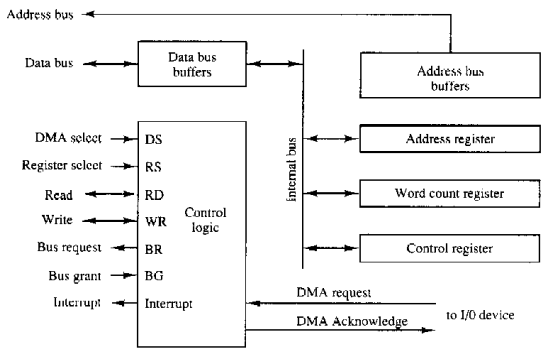


그림 11-17 DMA 제어기의 블럭도

RD(읽기)와 *WR*(쓰기)입력은 양방향으로 동작한다. *BG*(버스 승인)가 0일 때는 데이타 버스를 통하여 CPU와 DMA 레지스터 사이에 데이타를 주고받을 수 있고, *BG*(버스 승인)가 1일 때는 CPU가 버스를 포기하게 되므로 DMA가 직접 주소를 지정하고 *RD*, *WR* 제어를 활성화하여 메모리와 통신할 수 있다. 또

직접 메모리 접근 (DMA) (계속)

Slide 35

- CPU 의 DMA 초기화 순서
  1. 데이타가 존재하거나 (읽기) 저장될(쓰기) 메모리 블럭의 시작주소
  2. 메모리 블럭의 워드수를 나타내는 워드 카운트
  3. 읽거나 쓰기같은 전송모드를 지정하는 제어
  4. DMA 전송을 시작하게 하는 제어

- DMA 전송
  - DMA 구성 (그림 11-18)

직접 메모리 접근 (DMA) (계속)

Slide 36

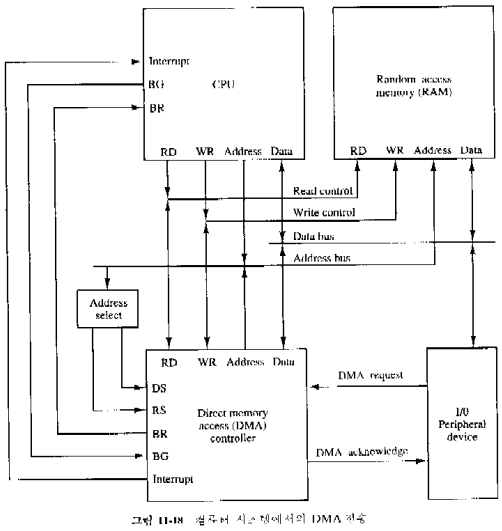


그림 11-18 컴퓨터 시스템에서의 DMA 전송

입출력 프로세서

- 입출력 전담 프로세서를 두어 입출력 장치와 직접적인 통신을 전담시킴
- 구성 (그림 11-19)

Slide 37

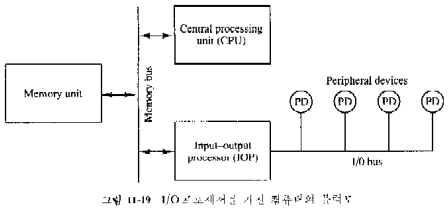


그림 11-19 I/O 프로세서를 가진 컴퓨터의 블럭도

- IOP 의 동작
  - DMA와는 달리 IOP 고유의 명령어를 fetch 실행할수 있음
  - 산술, 논리, 분기, 코드 변환등의 작업도 수행가능
- CPU-IOP 통신

입출력 프로세서 (계속)

Slide 38

– 통신의 흐름도 (그림 11-20)

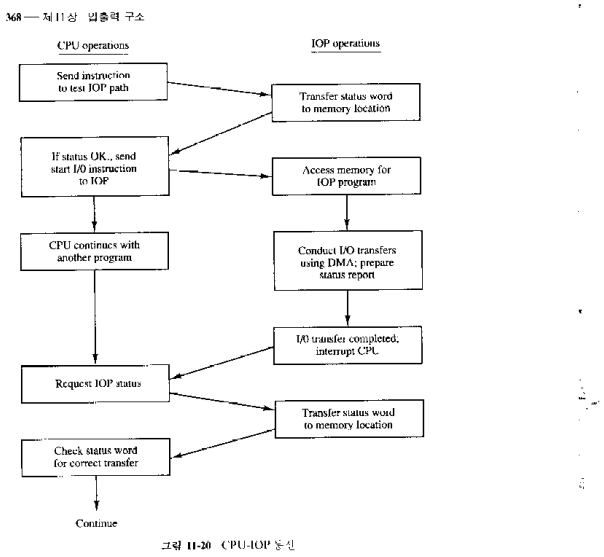


그림 11-20 CPU-IOP 통신

입출력 프로세서 (계속)

Slide 39

- Intel 8089 IOP
  - 50개의 명령어
  - 비트, 바이트, 16비트 워드 단위의 연산 가능
  - 논리, 산술, 및 조건 무조건 분기 동작, 서브루틴 및 점프 동작기능
  - 구성 (11-23)
  - CPU 와 8089 IOP 사이의 I/O 동작을 위한 메모리의 정보 (그림 11-24)

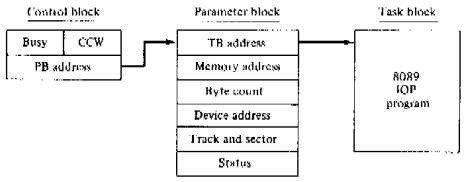


그림 11-24 인텔 8086/8089 마이크로 컴퓨터 시스템에서 I/O동작을 위한 메모리에서 정보의 위치

직렬 통신

- 전화선이나 통신회선을 통한 원격 데이타 송수신시
- 전화선을 통할때는 modem (modulation-demodulation) 필요
  - 아날로그 신호 $\Longleftrightarrow$디지탈 신호
- 통신 방식

Slide 40

  - 동기식 전송 →동기화 특수 시작종료 문자 사이에 대량의 정보전송
  - 비동기식 전송 →시작-정지 비트 사이에 한바이트 정보 전송
  - 에러 조사
    - 패리티 조사 →보통 CRC (cyclic redundancy check) 방식
    - 수신 문자 echo
  - 데이타 전송 방법
    - simplex : 단방향 통신 (라디오, TV 방송)
    - half-duplex : 양방향 , 특정순간 한방향
    - full-duplex : 동시 양방향 데이타 전송 가능

직렬 통신 (계속)

  - 통신선, 모뎀 및 다른 장비 →데이타 링크 (data link)
  - 데이타 링크상의 정보 전달 규칙 →프로토콜 (protocol) 이라함
- 문자 지향 프로토콜
  - ASCII 통신 제어 문자 (표 11-4)

Slide 41

표 11-4 ASCII통신 제어 문자

Code	Symbol	Meaning	Function
0010110	SYN	Synchronous idle	Establishes synchronism
0000001	SOH	Start of heading	Heading of block message
0000010	STX	Start of text	Precedes block of text
0000011	ETX	End of text	Terminates block of text
0000100	EOT	End of transmission	Concludes transmission
0000110	ACK	Acknowledge	Affirmative acknowledgement
0010101	NAK	Negative acknowledge	Negative acknowledgement
0000101	ENQ	Inquiry	Inquire if terminal is on
0010111	ETB	End of transmission block	End of block of data
0010000	DLE	Data link escape	Special control character

  - 문자지향 프로토콜의 메세지 형식 (그림 11-25)

직렬 통신 (계속)

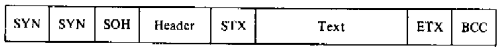


그림 11-25 문자지향 프로토콜의 전형적인 메시지 형식

- 전송의 예
  - * 단말 장치로부터 프로세서로 전송 (표 11-5)

Slide 42

표 11-5 단말 장치로부터 프로세서로의 전형적인 전송

Code	Symbol	Comments
0001 0110	SYN	First sync character
0001 0110	SYN	Second sync character
0000 0001	SOH	Start of heading
0101 0100	T	Address of terminal is T4
0011 0100	4	
0000 0010	STX	Start of text transmission
0101 0010		
0100 0101	request	Text sent is a request to respond with the balance of
.	balance	account number 1234
.	of account	
.	No. 1234	
1011 0011		
0011 0100		
1000 0011	ETX	End of text transmission
0111 0000	LRC	Longitudinal parity character

직렬 통신 (계속)

* 프로세서로 부터 단말장치로 전송 (표 11-6)

Slide 43

표 11-6 프로세서로부터 단말 장치로의 전형적인 전송

Code	Symbol	Comments
0001 0110	SYN	First sync character
0001 0110	SYN	Second sync character
1000 0110	ACK	Processor acknowledges previous message
0001 0110	SYN	Line is idling
.	.	
.	.	
.	.	
0001 0110	SYN	Line is idling
0000 0001	SOH	Start of heading
0101 0100	T	Address of terminal is T4
0011 0100	4	
0000 0010	STX	Start of text transmission
1100 0010		
1100 0001	balance	Text sent is a response from the computer giving the
.	is	balance of account
.	$100.00	
.		
.		
1011 0000		
1000 0011	ETX	End of text transmission
1101 0101	LRC	Longitudinal parity character

다. 첫 2개 문자는 헤딩 SOH와 T4의 단말 장치 주소를 가지고 있다. 텍스트 정보는 "balance is $100"이다. LRC가 계산되어 단말 장치에 보내어진다. 단말 장치가 NAK 문자로 응답하면 프로세서는 다시 이를 보낸다.

직렬 통신 (계속)

Slide 44

- 데이타 트랜스페런시 (data transparant)
  - * 문자가 아닌 이진정보 전송시
  - * 전송될 데이타에 통신제어 문자가 포함되면 문제 발생
  - * DLE (00010000) (data link escape) 비트를 통신제어 문자 이전에 삽입
  - * 수신부 에서
    - · DLE 조사 또 DLE 이면 뒤 DLE 는 정보, 앞 DLE 제거
    - · DLE 조사 DLE 가 아니면 통신제어 문자, 앞 DLE 제거

- 비트 지향 프로토콜
  - 프레임 (frame) 라 불리는 특별한 형식으로 구성됨 (그림 11-26)

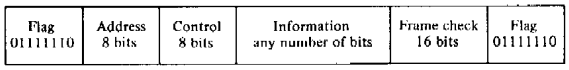


그림 11-26 비트지향 프로토콜의 프레임 형식

임은 8비트 플래그 01111110으로 시작되며 주소와 제어 시퀀스가 뒤따른다.

  - 종류

직렬 통신 (계속)

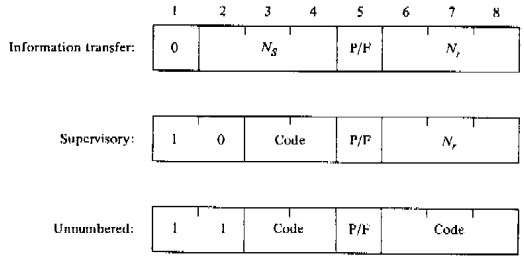
Slide 45

1. IBM에서 사용 →SDLC (synchronous data link control)
2. ISO 제정 →HDLC (high-level data link control)
3. ANSI에서 채택 →ADCCP (advanced data comm. control procedure)

- Flag 바이트로 시작 끝을 알림
- 연속되는 frame에는 하나의 flag 가 시작 끝을 알림
- 끝 flag 전의 16비트는 CRC 에러 점검 시퀀스
- 정보중에 flag 와 동일 정보가 있으면
  - * 송신부는 다섯개의 1후에 0을 삽입
  - * 수신부는 정보중에 다섯개의 1후의 0을 제거
    →정보중에 flag 는 없다
- 주소 필드는 수신부 주소
- 제어 필드의 구성 (그림 11-27)

직렬 통신 (계속)

Slide 46



N_s Send count P/F Poll/final
N_r Receive count Code Binary code

그림 11-27 비트지향 프로토콜의 제어 필드 형식