

디지털시스템과 마이크로 프로세서 설계 (VI)

金 明 恒*

요 약

마이크로 컴퓨터 입출력 방법을 논의한다. 세가지 입출력 방법으로 Programmed I/O 와 Interrupt 와 DMA 에 대해서 설명하고, 각 마이크로 프로세서의 입출력의 특징을 비교한다.

1. 서 론

컴퓨터에 있어서 내부연산 혹은 기억을 행하기 위한 자료의 입수 및 표시를 위하여 주변기기와의 입출력이 요구되며, 응용목적에 따라 적절한 방식을 채택하는 것이 최적 설계의 우선 요건이 된다.

입출력 방식은 응용목적에 따라 나눌 수 있으나 다음에서 설명하는 바가 그 기본이 된다.

입 출력 Device Selection

입출력 제어가 CPU의 Data Bus 에 연결되기 위해서는 Device Select 신호가 필요하다. 이 Device Select 신호를 발생시키는 방법에 따라 (가) Memory Mapped I/O 와 (나) I/O Mapped I/O 로 나누어진다.

2. 자료 처리 방법

CPU 가 입출력 자료처리에 관여하는 정도에 따라 (가) Programmed I/O (나) Interrupt I/O (다) Direct Memory Access (DMA) 로 구분할 수 있다. 일반적으로 (가) - (다)로 진행함에 따라 Hardware 가격은 증가하나 아래와 같이 System 의 효율은 상승한다.

2.1 Programmed I/O

가장 단순한 방법으로서 Hardware-Overhead 는 거의 없다.

이 방식은 입출력 Device 의 자료 대기시간이 전체 시스템의 효율을 극히 저하시키도록, 여러 가지 Device 를 Micro-Computer 가 제어해야 할 경우 혹은 빠른 자료전송을 요구하는 경우에는 사용할 수 없다.

자료의 이동방향이 출력인 경우에는, 출력 요구시

표 1

	Memory Mapped I/O	I/O Mapped I/O
정의	I/O는 Memory 의 일부로서 취급된다.	I/O는 Memory 와 분리되어 운영된다.
장점	I/O Operation 에 여러 가지 강력한 명령을 내릴 수 있다.	<ul style="list-style-type: none"> o 모든 Memory address 를 순수한 Memory 로 사용이 가능 o Decoding 이 단순해진다
예	Motorola 6800 Rockwell 6502	Intel 8080 / 85 Zilog Z 80 (Memory Mapped I/O 로 사용이 가능)

* 正會員 : 美國 Cornell 大 電氣工學科 教授 · 工博

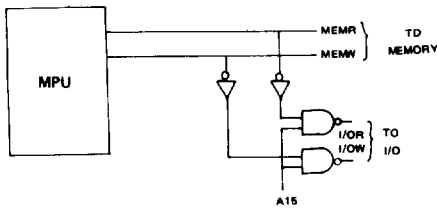


그림 1. Memory Mapped I/O

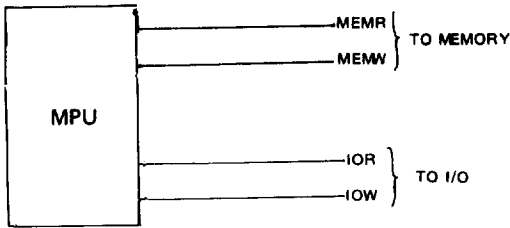


그림 2. I/O Mapped I/O

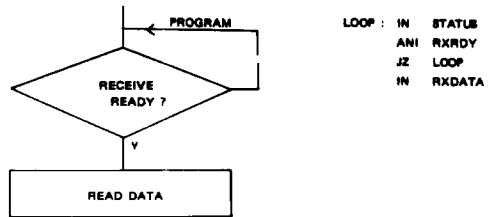
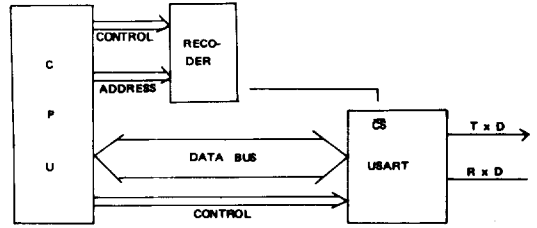


그림 4. Serial Interface -1

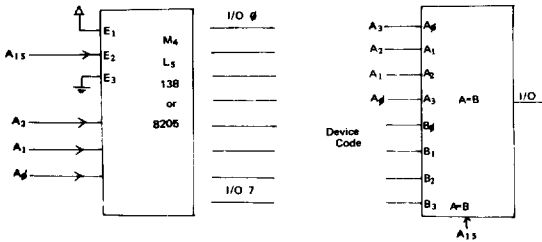


그림 3. Decoder 와 Comparator

2.2 Interrupt I/O

가) Interrupt의 필요성

Programmed I/O의 자료 대기시간을 줄이기 위하여, 자료가 발생될 때 마다 CPU에 그 발생을 알려 주어 자료 처리를 하는 것이 바람직하다. 이에 따라 Hardware cost는 증가하나 CPU Overhead는 감소한다 (표 2 참조)

나) Interrupt Process

Interrupt 구조는 Micro-Computer 마다 다르나 일반적으로 다음과 같은 Process를 거친다.

1. 현재상태의 보존
2. Service를 요구한 Device 확인
3. 해당 Device Service
4. 보존된 상태의 복원
5. Interrupt된 Process로 CPU Control 이전

1, 4의 Process는 Stack을 사용하는 것이 일반적이다. 빠른 Interrupt service를 요하는 경우 1, 4의 "DEAD TIME"이 가장 적은 Processor 선택이 바람직하다.

다) Multiple Interrupt Process

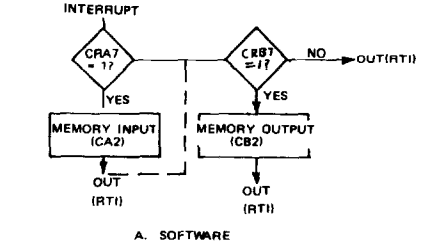
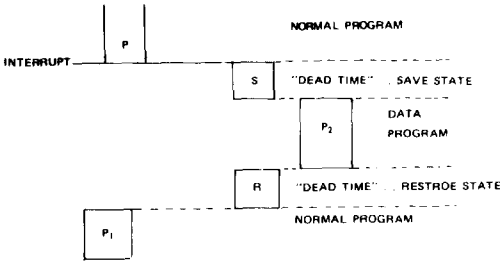
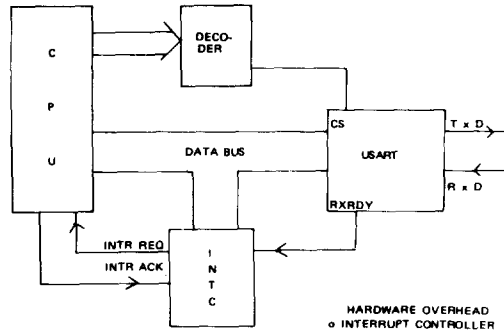
여러개의 입출력 Device가 Interrupt process를 요구하는 경우 다음과 같은 처리방법들이 있다.

1. Single Interrupt / Software Polling
2. Multiple Interrupt / Daisy Chain

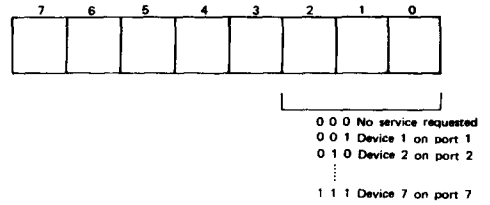
간이 CPU 자체에서 알 수 있으므로 Programmed I/O가 가능한 경우가 많다. 그러나 입력인 경우에서 발생 즉시 자료처리가 안될 수 있으므로 거의 사용하지 않는다.

표 2. CPU Overhead

Programmed	Interrupt	DMA
100 %	1 %	0.01 %
자료전송대기시간이 대부분임.	자료전송처리 Program 소요시간	자료전송시 1Cycle 당 1Cycle 이소모
약 100 CH/SEC	약 10 ⁴ CH/SEC	약 10 ⁶ CH/SEC



A. SOFTWARE



B. STATUS BYTE 의 예

그림 5 . Serial Interface - II

- 3. Multiple Interrupt / Priority Vectored
- 4. Multiple Line Interrupt

각각의 방법중 어떠한 방법을 선택하느냐 하는 것은 입출력의 자료 처리속도에 달려 있다.

1. Single interrupt / Software Polling

그림 6 과 같이 interrupt 가 발생할 경우 여러개의 Device 의 Request 를 Polling 한다. 대체로 Timer에 의해서 Interrupt Request 를 행하는 경우가 많다. 이 방법은 Hardware cost 는 가장 저렴하나 CPU Overhead 가 크다.

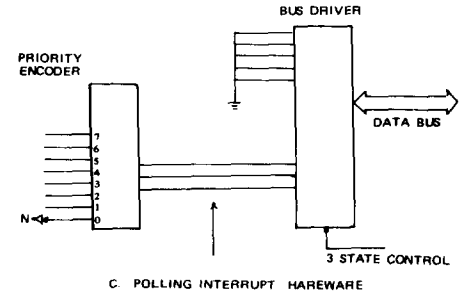
2. Daisy Chain

그림 7 과 같이 배열된 순서에 따라 우선 순위가 결정되어 Interrupt Ack 에 의해 Device Vector 를 Data Bus 에 실어 Service 를 받는다.

3. Priority Vectored Interrupt

그림 8 과 같이 Interrupt Ack 에 의해 미리 결정된 우선 순위에 따라 Service Address 를 발생한다. 동시에 여러 Device 가 Interrupt Service 를 요구할 경우를 위하여 미리 Priority 를 정하며 경우에 따라 interrupt 를 제어하기 위하여 Mask Logic 를 사용한다.

그림 2 - 10 은 Intel 8259 FIC 의 block Diagram 이다.



C. POLLING INTERRUPT HAREWARE

그림 6 . Interrupt Polling

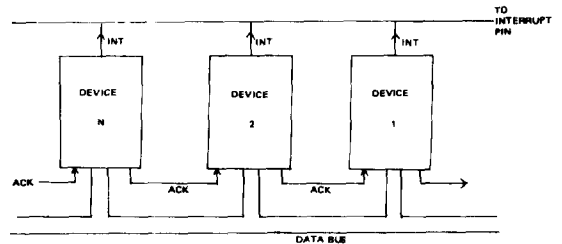


그림 7 . Daisy Chain

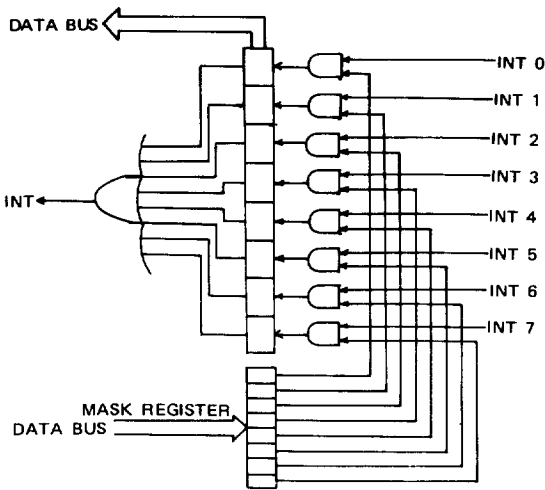


그림 8 . PIC 논리 회로

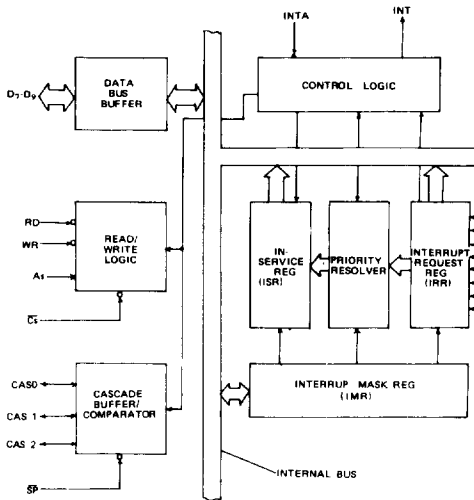


그림 9 . Intel 8259 PIC

라) 8085 Interrupt Structure

Intel 사의 8085 Interrupt 압력은 5개가 있다.

1. INTR..... 8080 의 Intel 와 동일한 기능을 갖는다. Intra 시간에 Data Bus 에 RST 혹은 Call Instruction 을 실행해서 Service Routine Branching 을 행한다.
2. RST 5.5, 6.5, 7.5 Hardware Interrupt 로서 Intra 를 필요하지 않고 미리 정해진 Service Routine 으로 Branch 된다. 5.5 와 6.5 는 level-input 이며 7.5 는 edge-input 이다.

각 Interrupt 는 SIM 명령에 의해 Mask 된다.

3. TRAP RST 와 같으나 Non - Maskable 이므로 Power -fail 혹은 Bus Error와 같은 경우에 사용된다.
4. Priority TRAP, RST 7.5, BST 5.5 의 순서로 우선 순위가 주어진다. 8085 Micro-Processor 는 다른 8 Bit Micro-Processor 에 비하여 다양한 In-terrupt 기능을 가지므로 Small System 에 사용하기에 적합하다.

마) Motorola 6800 Interrupt Structure

기본적으로 6800은 Polled Interrupt 구조로 되어 있으며 다음과 같은 Processing 을 거친다.

- 1) Interrupt 이 발생하면 현재 Instruction 수행을 마친 후 (Last Cycle 이면 다음 Instruction) FLAG , ACC, INDEX, PC 를 Stack 에 저장한다.
- 2) FFF 8 및 9 에 저장된 Pointer 의 위치에서 PC 를 Load 한다.
- 3) Status Bit 를 Poll 하여 Device 를 선택하여 Service 한다.
- 4) Stack 에 저장된 State Vector 를 회복하여 Normal Program 을 수행한다.

6800 CPU 를 사용하여 Vectored Interrupt 를 행하기 위해서는 그림 10 과 같은 Hardware 를 구성해야 한다. 즉 FFF 0 에서 FFFF 가 Address 될 때는 MUX 를 통하여 Priority Encoder 가 ROM Address 입력이 되도록 하여 Service Routine Branch 를 수행할 수 있다.

6800 에서 8085 의 TRAP 에 해당되는 NMI interrupt 가 있으며 FFFC 에서 FFFD 에 pointer 가 저장되어 있다.

바) Moster 650 X Interrupt Structure

650 X Series 는 6800 과 거의 같으나 다음과 같이 약간의 차이점이 있다.

1. 24 pin CPU 는 NMI Pin 을 다른 용도로 사용한다. 6504 (A 12) , 6505 (Ready)
2. 6800 의 State Vector Saving 이 없다.
3. DEBUG 을 위하여 SWI 대신 BRK 명령을 사용한다.

사) Z - 80 Interrupt Structure

Z 80 은 8080 에 비하여 3 가지 Mode 를 보유하고 있다.

1. 8080 Mode
2. Fixed Mode 8085 의 RST 와 동일

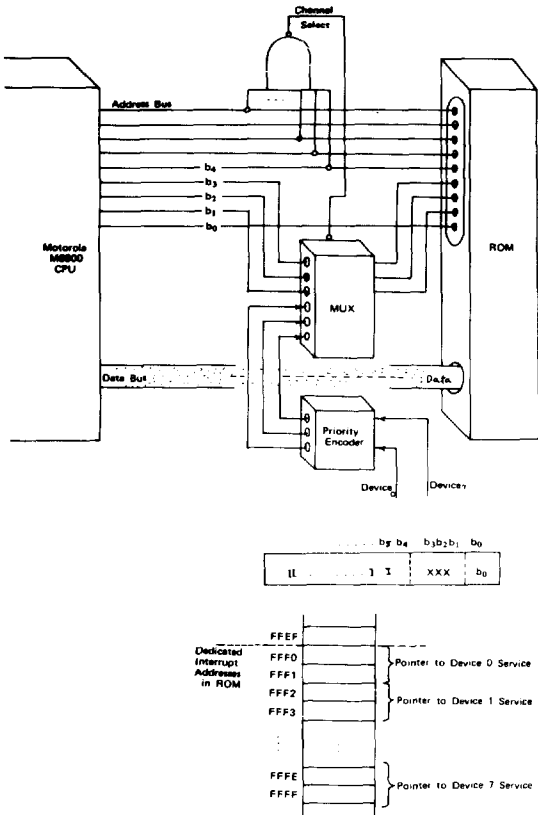


그림10. 6800 Vectored Interrupt

(60 M)

3. Z 80 Mode Vector Address (1 Byte) 만 받는다.

2.3 Direct Memory Access (DMA)

Interrupt 에 의하여 Programmed I/O 의 문제가 되는 자료대기시간을 줄일 수 있으나 자료처리 시간을 줄일 수 없다. DISK 와 CRT Refresh 와 같이 신속한 자료이용이 요구되는 경우 Interrupt 방식에 의한 자료처리는 (Processor 에 따라 다르나) 15CH /sec 를 넘어설 수 없다. 그러나 Memory 의 Cycle Time 이 Microsec 라 할 때 Memory 자체는 IMCH /Sec 의 자료전송을 할 수 있다.

따라서, 입출력 장치와 Memory 간에 직접 자료전송을 행하게 하는 방법이 요구되며, 이를 DMA 라 한다.

1. 기본적인 DMA 요구사항

DMA 에는 다음과 같은 기능이 요구된다.

- 1) DMA Address , Length , Mode 결정

- 2) DMA Handshake
- 3) CUP BUS Floating
- 4) CPU Signal Generation

각각의 자세한 설명은 실제로 Intel 사의 8257 DMAC 를 통하여 하기로 한다.

2. 8257 DMAC (그림 11)

1) 8257 DMAC 에는 4 개의 DMA Channel 이 있다. 각 Channel 의 MODE (Read , Write , Verify) 와 Memory 및 Address 는 미리 Program 된다.

2) 외부 Device 로부터 DMA Request 를 받으면 DMAC 는 8085 / 85 CPU 의 Hold 입력에 DMA Request 를 보내면 CPU 는 Hold Ack 를 DMAC 에 보내고 Hold 상태에 들어간다 (Bus Floating) DMAC 는 Hold Ack 를 받으면 Requesting Device 에 DMA Ack 를 보낸다.

3) System Bus 의 제어를 DMAC 가 행하여 (이를 위하여 Address Bus 도 tri -state 가 되어야 한다) 미리 Program 된 Memory Address 로부터 주어진 길이만큼 자료를 읽거나 쓴다.

4) DMAC 는 Host CPU 의 모든 signal 을 만들어야 하므로 반드시 동일한 회사제품을 사용해야 한다.

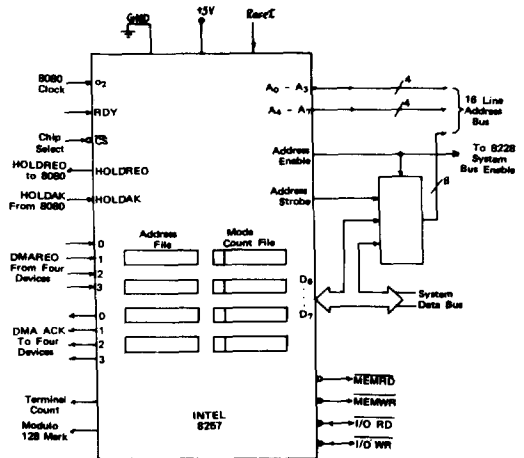


그림11. Intel 8257 Block Diagram

3. 입력출 시간 (I/O Timing)

적절한 Micro-Processor 설계를 위하여 I/O Timing 분석을 통하여 Access 방법을 결정해야한다. 대부분의 I/O Timing 의 문제는 외부 Device 과 Micro-Computer 사이에 Array Data 처리에서 발생

한다. 일반적으로 이러한 자료처리에는 다음과 같은 시간이 변수로서 작용한다. (그림 2-17, 2-18)

1. T (Device)Device 의 자료처리 간격
2. T (PRE)Device 의 Ready 신호를 받고 다음 자료 준비시간
3. T (Micro) Device 를 Service 하는 시간의 합 여기에는 T (PRE) 와 자료Test 기간의 포함된다.

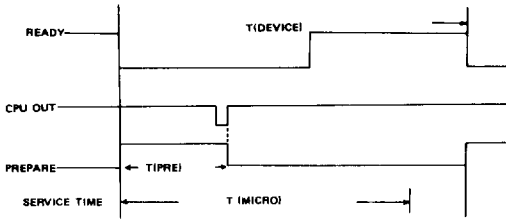


그림12 . 기본적인 시간변수

Under Flow를 피하기 위해서는

$$T (PRE) MAX \quad T (Device) MIN \quad \dots\dots \text{필요조건}$$

$$T (Micro) MAX \quad T (Device) MIN \quad \dots\dots \text{충분조건}$$

이 만족되어야 한다.

참 고 문 헌

- [1] Intel, 8080 Microcomputer Systems User's Manual, 1975.
- [2] Zilog, Z80-CPU Technical Manual, 1978.
- [3] Motorola, MC 6800 Manual, 1977.
- [4] Hilburn, J. and P. Julich, Microcomputer/ Microprocessor: Hardware, Software, and Applications, Englewood. Cliffs, N.J., Prentice Hall, 1976.
- [5] Osborn, A. An Introduction to Microcomputers, vol. 1: Basic Concepts. Berkeley, Calif. Adam Osborn and Associate, 1976.
- [6] Mick, J and J. Brick, Bit-Slice Microprocessor Design, New York, NY: McGraw-Hill Book Co. 1980.
- [7] 김명환, 석민수, "디지털 시스템과 C. P. U. 설계, (산학협동하기단기강좌), 한국과학기술원, 1982.
- [8] 김명환, "디지털 시스템과 마이크로 프로세서 설계: V Bit Slice Microprocessor", 전기학회지, 제 31권, 1982.11.

新 規 加 入 者 名 單

(1982.11.1 ~ 11.30) : 31名 (正:25名 學生: 6名)

區分	會 員 番 號	姓 名	所 屬	區分	會 員 番 號	姓 名	所 屬	區分	會 員 番 號	姓 名	所 屬
正	575493	李康玩	한국전력공사	正	045028	高政秀	새한설계콘설탄트	正	793042	韓連錫	국립중앙직업훈련원
"	634071	趙京海	"	"	094046	權寧秀	"	"	392024	孫有植	통영수산전문대
"	793040	韓秉誠	전 북 대	"	122560	金基昇	"	"	122565	金永文	경 남 대
"	122559	金庚緒	서 울 대	"	122561	金明秀	"	學生	423044	申正鎬	충 남 대
"	423043	申小澈	"	"	122562	金 永	"	"	294009	邊煥弼	광 운 공 대
"	581002	印熙植	"	"	453026	梁成一	"	"	844042	黃仁範	"
"	620136	鄭勝基	"	"	575494	李秀榮	"	"	634072	趙盛桓	"
"	620137	鄭義相	"	"	575495	李永洙	"	"	122564	金光洙	"
"	620138	鄭且根	"	"	586016	任大彬	"	"	018057	姜聖卓	"
"	793041	韓己洙	"	"	280019	裴尙得	승전대학원				
"	225002	明聖鎬	"	"	122563	金洪必	한 양 대				