TDX-1 시스템의 I/O Device 処理方式에 관한 小考

黃勝會·金學星·吳昌煥 / 프로세서開発室

〈要　約〉

본고는 교환기에 사용되는 각종 I/O device를 분류한 후 일반적인 I/O device의 처리방식을 서술하였으며, ANSI에서 개발한 이들 방식의 implementation technique를 TDX－1 시스템의 storage architecture에 적용시키는 방안을 제시하였다.

Ⅰ. 서　론

교환기에서 사용하는 각종 I/O device들은 man-machine 및 machine사이의 정보 전달과 시스템 내의 프로그램 또는 데이터를 저장하는 storage의 기능 등을 제공하여 프로세서 시스템 내의 I/O 서브 시스템을 구성한다. 또한 이 I/O서브 시스템의 구성 형태는 전 시스템의 효율적인 동작과 구성가격에 큰 영향을 주게 되므로 I/O device 처리 방식의 결정과 그에 따른 implementation technique은 시스템 구성시 고려해야 할 중요한 사항들이다.

본고는 I/O device 처리의 일반적 방식을 소개한 후 ANSI에서 개발한 이들 방식의 implementation technique를 TDX－1 시스템의 storage architecture에 적용시키는 방안의 내용을 다음과 같이 구성하였다. 즉, 2장에서는 각종의 I/O device를 분류한 후 I/O 시스템의 구성을 I/O채널, control unit, device의 3단계로 나누어 각 단계에서의 처리 방식을 개별적으로 설명하였으며 이후 3장은 ANSI의 I/O interfacing technique를 소개하고 4장에서는 이들의 적용 방안을 제시하였다.

Ⅱ. I／O Device의 분류 및 처리방식

1. I/O Device의 분류

I/O device의 분류는 기기의 자체 특성 또는 기기를 요구하는 각 job에 이들을 할당하는 방식에 따라 각각 다음과 같이 분류된다.

가. 기기의 특성에 의한 분류

CRT 또는 TTY와 같은 전송자료의 입출력 기기와 파일저장용 기기로 분류되며 파일 저장용 기기는 다시 액세스 방식 및 액세스 시간에

의해서 순차 액세스 기기와 직접 액세스 기기로 세분화 된다.

나. Allocation 방식에 의한 분류

• Dedicated Device : 테이프 또는 프린터와 같이 초기 가동시에 많은 시간이 소요되거나 수행되고 있는 여러개의 job들이 공유할 수 없는 기기

• Shared Device : 대개의 직접 액세스 기억장치들은 여러개의 job들에 의해서 공유되며, 각 job에 할당하는 방식은 I/O traffic controller 및 scheduler와 같은 S/W에 의해서 결정되거나 controller unit와 같은 H/W에 의해서 결정된다.

• Virtual Device : Dedicated되어진 기기가 spooling프로그램에 의해서 shared device와같은 개념으로 동작하는 기기

2. I/O Device의 처리방식

전절에서 설명한 각각의 I/O device들을 효과적으로 동작시키기 위해서는 기기의 점유상태를 확인한 후 이들을 각 job에 allocation 및 deallocation을 하는 I/O device 처리 방식이 결정되어야 한다.

또한 multitasking 또는 single user 용으로 구분되는 I/O device의 용도와 기기의 특성 등을 고려 대상으로 하는 기기의 처리 방식은 I/O 채널, control unit, device의 3단계로 구성되는 I/O시스팀의 구성형태를 결정하게 된다.

가. I/O 채널 및 제어 장치의 처리방식

I/O시스팀의 구성 형태 중 최상위의 단계인 채널의 유형은 고속 I/O에 적용할 수 있는 선택 채널, 카드리이더 및 프린터와 같은 저속으로 동작하는 기기에 해당하는 multiplexer channel, 고속으로 동작하는 기기를 사용하는 multitasking용의 block multiplexer channel 등으로 구성된다.

시스팀의 구성시 채널과 제어장치의 수는 구성가격을 고려하여 관련된 I/O device들의 수에 비해 일반적으로 적게 할당된다. 따라서 이들 각각을 대응시켜 경로를 구성한 후 기기를 동작시키는 방식에 문제점들이 생기게 되며 이에 대처하는 일반적인 H/W 방법은 다음과 같다.

1) I/O device의 독립적인 동작 방법

뒤에서 설명할 ANSI의 인텔리전트 인터페이스와 같이 인텔리전트 기능을 갖고 있는 주변기기를 사용할 경우 이 기기들은 채널 또는 제어장치의 계속적인 도움없이도 I/O 동작을 수행할 능력을 갖고 있다.

예를들면 disk seek 동작과 같이 가장 긴 시간이 소요되는 기기의 동작을 수행시킨 동안 해당 채널과 제어장치를 사용하여 다른 기기를 액세스할 수 있는 것이다.

이때 고려해야 할 사항은 각 단계간의 경로 상태를 감시하는 기능의 필요성이며 이러한 방법은 채널내의 채널 상태어를 사용하여 해결한다.

2) Device buffering

이 방법은 기기 또는 제어 장치내에 버퍼 기능을 두어 정해진 버퍼 크기에 해당하는 정보가 입력된 후에 경로를 이용하는 방법으로 보통 저속으로 동작하는 기기들에 적용된다.

3) Multiple path

각 I/O device에 액세스하는 경로를 여러개로 하여 액세스하려는 경로의 busy시에는 다른 경로를 이용할 수 있는 방식이며, 이때 각 경로의 사용 가능성은 device, 제어장치, 채널의 상태 블럭을 관장하는 I/O traffic controller 에 의해서 결정된 후에 각 경로의 연결이 형성 된다.

이 방식은 버스상의 신뢰도를 높여줄 수 있다는 장점도 갖고 있다.

앞에서 설명한 방식들 외에도 multiprogramming을 위한 H/W 방식인 block multiplexer채널을 사용할 수도 있으나 이것은 채널에서 빠른 device switching을 해야되므로 고속으로 동작하는 프로세서를 이용해야 된다는 단점을 갖고 있다.

나. I/O Device의 구성 방식

I/O시스팀의 신뢰도와 정보의 안전성은 기기 자체의 신뢰도와 기기들의 구성 방식에 의해 큰 영향을 받게 되므로 기기의 선정 및 구성 방식의 결정은 시스팀의 구성시 중요한 사항들이 된다.

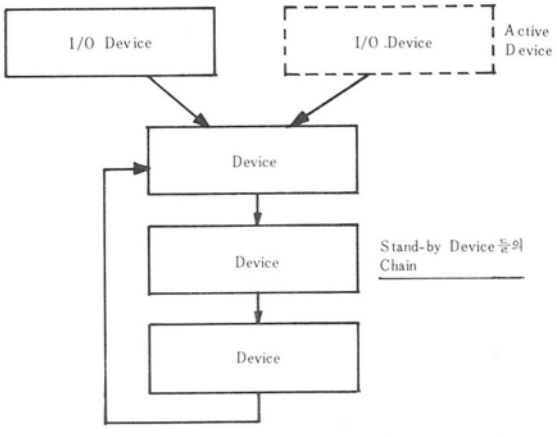
우선 기기들의 구성 방식은 크게 active/stand-by와 continuation의 두가지 형태로 분류할 수 있다.

1) Active/Stand-By의 구성

〈그림 1〉에 보인바와 같이 active/stand-by의 구성은 다음과 같은 특성을 갖고 있다.

Stand-by device는 1개 이상의 active device에 대해 서어비스를 하며, stand-by device가 active의 상태를 take-over하는 경우는 active device에 fault가 발생할 때이며 active가 busy일 경우의 입력 데이터는 queue를 형성하고 take-over 현상은 발생하지 않는다.

Stand-by device에 fault가 발생할 경우는 자동적으로 다음 기기가 active 상태를 take-over하게 된다.

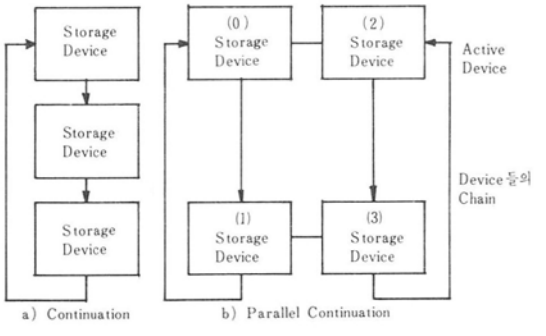


〈그림 1〉 Chain의 형태를 가진 Active/Stand-by의 구성

2) Continuation의 구성

과금 또는 통계용 자료와 같이 많은 양의 자료를 저장할 경우에 사용하는 방식으로 파일저장용 기기에만 적용되며 1개의 active device 단위로 chain이 구성된다.

Continuation의 구성은 〈그림 2〉와 같이 일반적인 continuation과 parallel continuation의방식으로 분류된다.



〈그림2〉 Continuation과 Parallel Continuation의 구성

Continuation의 구성을 가지는 기기들의 동작은 다음과 같다.

Active storage device가 full인 경우의 입력 자료는 다음의 continuation device로 계속해서 입력되며 마지막 순위의 기기는 처음 기기를 continuation device로 한다. Parallel continuation은 높은 신뢰도를 요구하는 storage system의 구성에 적합하며 동작의 예를들면 다음과 같다.

입력 자료는 parallel device (0), (2)로 동시에 입력되며 active device가 동작중에 full 이 되거나 자체에 fault가 발생할 경우는 다음의 continuation device들이 pair로 동작하게 된다. 이후 device(3)에 fault가 발생하면 이미 다음 continuation device의 pair는 사용할 수 없는 상태이므로 자료는 device (1)으로만 입력되게 된다.

Ⅲ. ANSI의 Interfacing Technique

앞에서 설명한 3단계의 구조를 가진 기존의 I/O시스템에서 기기의 처리는 호스트 컴퓨터의 오퍼레이팅 시스템 또는 제어장치 내의 peripheral-driving routine에 의해서 수행되므로 기기의 변경시에는 그 상위 단계의 변경을 초래하여 시스템 전체의 재구성을 필요로 하게 된다.

이러한 불편을 덜기 위한 방법으로서 인텔리전트 H/W를 주변기기에 장착하여 종래의 peripheral-driving routine을 high-level command로 대체하는 방안들과 이에 따른 주변기기의 인터페이스를 표준화하는 방안들이 발표되었다.

Intelligent를 갖고 있는 주변기기를 인터페이스하는 기술은 Control Data Corp.에서 개발한 ISI(Intelligent Standard Interface), ANSI X3T9.3의 IPI(Intelligent Peripheral Interface), SASI Interface를 개량한 ANSI X3T9.2의 SCSI(Small Computer System Interface) 등이 현재 사용되는 대표적인 방법들이다.

ISI와 IPI의 방식은 master-slave의 처리방식이며 ISI가 소형 컴퓨터 시스팀용인데 반하여 IPI는 IBM의 시스팀 360/370 Mainframe computer의 I/O channel architecture 에서 실현되어 그후 mainframe의 표준화된 I/O 인

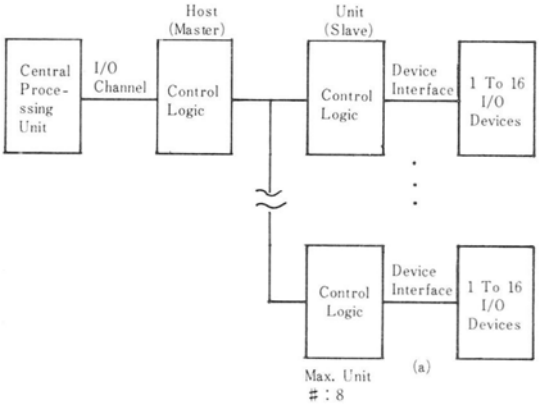
터페이스로서 사용되는 방식이다.

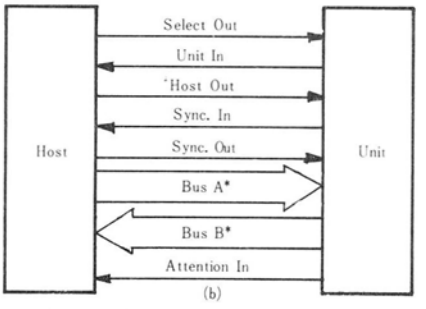
ISI는 아직 공인성이 미흡한 상태이므로 여기서는 ANSI의 표준 인터페이스인 IPI와 SCSI에 관해서 언급하기로 한다.

1. Intelligent Peripheral Interface

ANSI X3T9 subcommittee의 group인 X3T9. 3는 mainframe channel interface 용으로 intelligent peripheral interface를 개발하였는데, 이것은 수행할 채널 인터페이스의 기능을 physical operation과 logical operation으로 분리한 layered architecture를 가능하게 하여 주변기기들의 변경을 용이하게 하였다.

처리 방식은 기본적인 master-slave의 구조를 가지며, 〈그림 3〉에서와 같이 5개의 control line을 이용한 state transition과 interlocked handshaking에 의하여 정보 전송이 가능하게 한다. 이때 최대 전송 속도는 single byte 및 double byte전송시에 각각 1.5MByte /s와 3 MByte/s가 된다.





(*: 8Bits+1 Parity Bit의 Bus Cable이며, Double-Byte 전송시는 양방향Mode로 동작)

〈그림 3〉 Master-Slave의 구조와 Control Signal

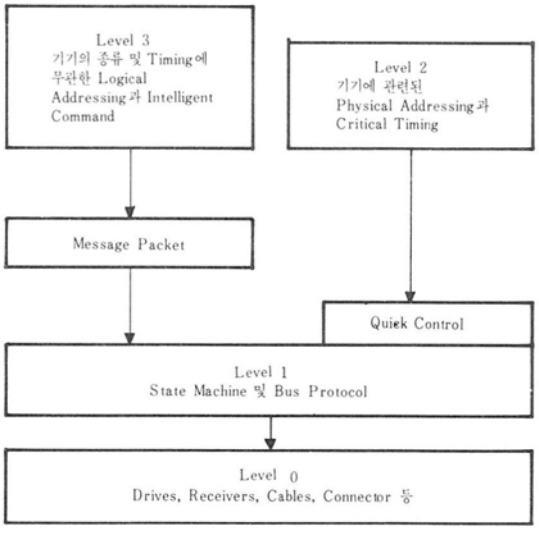
가. Interface Layer

IPI의 interface layer는 〈그림 4〉에서 보인 바와 같이 ISO의 OSI(Open System Interconnection) Model(참고문헌 Designer's Reference 참조)을 응용하여 구성하였다.

Highest level인 Level 3는 intelligent command structure를 사용하여 주변기기들을 처리할 수 있다. 즉. 기기 및 timing에 무관한 logical addressing command, controller에 부착된 기기들의 특성과 용량을 확인할 수 있는 attribute command, format command, normal read, write 및 copy와 같은 combination command 등의 high-level command들은 host, control unit-level에서의 시스템 구성을 기기와는 무관하게 할 수 있도록 한다.

Layer 2는 storage location의 physical addressing, 기기와 controller간의 정보 전송시 요구되는 timing등 intelligent interface 의 부분을 담당한다. 따라서 Level 2는 10MByte/s 정도의 전송 속도시에 요구되는 critical timing을 기기의 인터페이스에 제공하는 장점을 갖고 있다.

Level 1은 control line을 사용한 bus state의 transition과 state의 상태등 bus 및 timing의 프로토콜을 담당하며, Level 0는 host 와 주변기기간의 physical connection을 위한 H/W로 구성된다.



〈그림 4〉 IPI-Interface Layer의 구성

〈그림 4〉에 보인바와 같이 IPI의 intelligence는 command와 response의 message packet을 사용하여 Level 0, 1과 통신을 하는 Level 3에 의해서 결정되며, Level 2는 microcoded instruction을 decode하는 기능을 가진 physical interface를 통하여 quick control이 가능하게 한다.

나. IPI Bus의 동작방법

IPI bus의 동작은 〈그림 3〉 (b)와 같이 attention을 제외한 5개 제어 신호들의 변화에 따른 state transition에 의해서 수행된다.

우선 host는 Bus A에 3bit address를 실은 후 select-out line을 assert하여 unit selection state를 수행한다. 이에 대한 unit의 status response는 unit가 자신의 status를 Bus B에 실은 후 unit-in line을 assert하는 unit acknowledge state에 의해서 수행되며, 이후 host는 Bus A에 command를 실은 후 host-out line을 assert하며 ready-to-transfer의 state로 들어간다. 이후의 정보 전송은 sync-in, sync-out에 의한 handshake에 의해서 진행된다.

IPI의 operating states는 〈表 1〉 같이 실제 정보 전송에 필요한 state는 14개가 해당한다. 5개 제어신호에 의한 32개의 state중 12개는 reserved, 6개는 reset unit용으로 사용되어 14개의 state만이 bus 동작시에 사용된다.

Signal States	Mnemonic	Definition
0 0 0 0 0	Idle	Interface is in Idle
0 0 1 0 0	REQINT	Request Interrupt
0 1 0 0 0	DESEL	Deselect
0 1 1 0 0	INTACK	Acknowledge
1 0 0 0 0	SELECTED	Selected
1 0 1 0 0	UNITEND	Unit Ends Operation
1 1 0 0 0	UNITACK	Unit Acknowledge
1 1 0 0 1	BUSCMD	Bus Command
1 1 0 1 0	HOSTEND	Host Ends Operation
1 1 0 1 1	BUSACK	Bus Acknowledge
1 1 1 0 0	XFRRDY	Ready to Transfer
1 1 1 0 1	XFREND	End of Transfer
1 1 1 1 0	XFRST	Start of Transfer
1 1 1 1 1	XERRES	Respond to Transfer

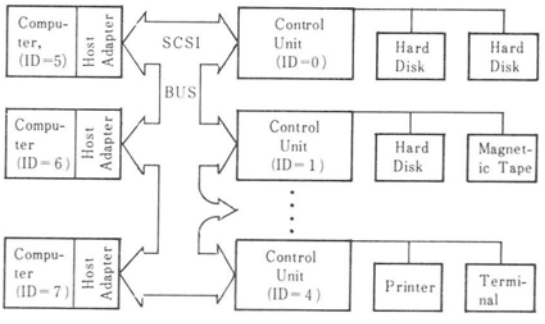
Sync Out
Sync In
Host Out
Unit In
Select Out

〈表 1〉 IPI의 Operating States

2. Small Computer System Interface

ANSI X3T9 subcommittee의 group인 X 3 T9. 2는 SASI(Shugart Associates System Interface)의 기능에 멀티 프로세서의 기능을 첨부하고 high-level command를 표준화 하여 SCSI(Small Computer System Interface)를 개발하였다. 따라서 IPI의 Layer 3 command들과 같은 성격을 갖는 SCSI의 high level command를 사용하여 host의 오퍼레이팅 시스템을 개발하면 인터페이스 버스에 연결된 주변기기들의 변화시에도 호스트의 S/W 및 H/W 설계에는 영향을 미치지 않는다.

〈그림 5〉에 보인바와 같이 SCSI Interface는 일반적 용도의 local I/O bus로서 priority arbitration 방식을 사용하여 멀티 프로세서의 동작이 가능하게 한다. 이때 동일 버스상에 연결될 수 있는 host 및 제어장치의 수는 데이터 버스의 비트 수와 같은 8개이다.



〈그림 5〉 Multiple Host들의 주변기기들을 공유하는 SCSI Bus의 구성

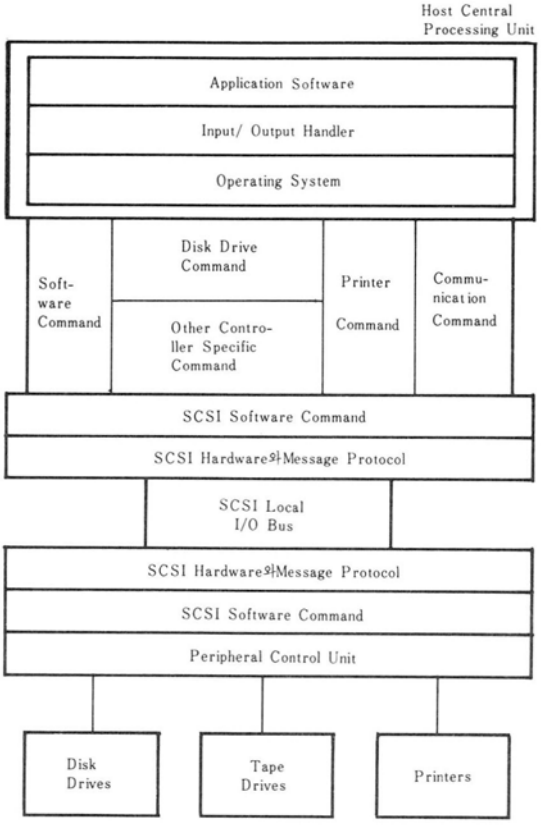
이러한 arbitration의 기능은 제어장치가 명령을 해석한 후 버스를 사용하지 않는 기능의 수행시에는 버스의 절체를 가능하게 하여 시스템의 throughput을 증진시키는 역할을 하지만 다음과 같이 bus device들간의 Physical path를 management하는 기능을 필요로 하게 된다.

즉 physical path를 management하는 방식으로 SCSI는 message packet을 이용하는 initiator-target 간의 통신방식을 사용한다. Physical path를 형성하기 위해서 initiator와 target은 bus address와 target control unit이 address할 주변기기의 logical unit number를 포함하는 identify message를 사용한다.

또한 target은 bus로 부터 절체와 재 연결의 기능을 수행하기 위해 각각에 해당하는 메시지를 initator에게 전달한다.

가. SCSI의 Interface Layer와 Bus의 동작 방법

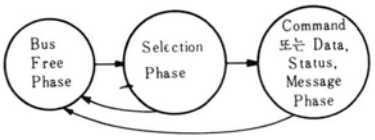
IPI의 Level 3에 해당하는 high-level command structure를 갖고 있는 SCSI의 interface layer는 개략적으로 〈그림 6〉과 같으며 여기서 physical interface는 H/W가 담당하고 있는 것을 알 수 있다.



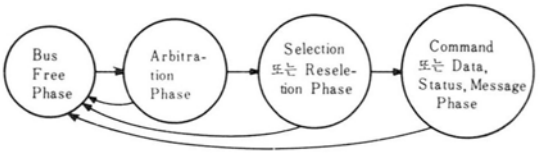
〈그림 6〉 SCSI Interface Layer의 구성

SCSI Bus의 동작방법은 single host system시와 multi-host의 arbitration 기능을 갖고 있는 경우 각각 〈그림 7〉의 (a), (b)와 같이 다른 phase sequence를 갖는다.

Selection phase 이후 information transfer phase에서 사용되는 제어신호들은 C/D, I/O, MSG 등이 해당되며 host와의 정보 전송은 acknowledge와 request에 의한 handshake 방식을 사용한다. 또한 SCSI local I/O bus를 통



(a) Single-Host SCSI의 Phase Sequence



(b) Multi-Host SCSI의 Phase Sequence

〈그림 7〉 SCSI Bus의 Phase Sequence

해 message packet으로 전달되는 command format은 〈그림 8〉과 같다.

Byte	Bit 7	Bit 6	5	4	3	2	1	0
00	Group Code			OP Code				
01	LUN			MSB Logical Block Address				
02	Logical Block Address							
03	LSB Logical Block Address							
04	Number of Block							
05	VU	VU	R	R	R	R	Flag RQST	Link

LUN :Logical Unit Number
MSB :Most Significant Bit
VU:Vendor Unique
R :Reserved

〈그림 8〉 Command Format

Group command는 0 – 7까지 분류되며, Group 0 command가 6바이트로 구성된 common command로 사용된다.

그 이외의 group command는 common command의 확장시나 vendor unique한 기능 등이 요구될 때 사용된다.

Ⅳ. TDX – 1 System의 I/O Processor Configuration 및 특성

1. 일반적 교환기의 I/O Subsystem 기능 및 Device의 구성

일반적으로 교환기에서 요구하는 I/O 서브시

스팀의 기능은 크게 operation과 maintenance 기능으로 분류할 수 있다.

Operation의 기능은

- Man-machine communication 기능
- 프로세서 시스팀의 back up용 프로그램 및 데이터의 저장과 수정 기능
- 과금 및 통계자료의 저장 및 처리기능
- 서어비스 감시 및 데이터 채널의 기능 등으로 세분화 되며,

Maintenance의 기능은

- 경보와 고장보고를 처리하는 기능
- 고장 추적의 기능
- 시스팀의 보수 및 시험 기능
- Inspection의 기능

등으로 다시 분류된다.

앞에서 분류한 각각의 기능을 수행하기 위하여 device block은 I/O device (H/W), nterface unit (H/W 및 해당 기능을 수행하기 위한 S/W로 구성되며 I/O device의 구성 방식은 Ⅱ. 1. 나. 절에서 설명한 바와 같이 TTY, CRT 등의 기기들은 단일 chain의 active/stand-by 형태로 구성하고 storage device의 경우에는 저장되는 정보의 양과 중요도에 따라 〈그림2〉의 continuation 또는 parallel continuation 방식으로 구성한다.

2. TDX－1 System의 I/O Device 처리방식과 특성

TDX－1 시스팀의 I/O device 처리 기능은 프로세서 구조상 D-level에 속하는 프로세서들이 수행한다.

이 D-level 프로세서들은 앞 절에서 설명한 I/O 서브시스팀의 각 기능들을 수행하기 위하여 해당 I/O device들을 제어하며 primary processor인 상위 T-level의 SAP (System Administration Processor)와 D-bus를 이용한 bit oriented serial 통신을 하여 교환에 필요한 기능들을 제공하고 있다.

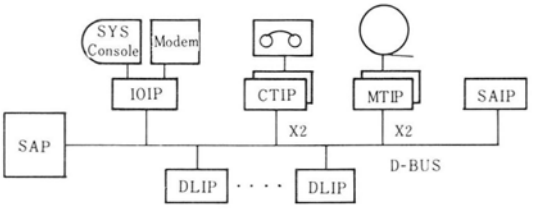
I/O device들에 관련된 데이터들의 처리는 SAP의 응용 프로그램에서 처리되며 I/O device들은 별도의 D-level 프로세서에 의해 제어되므로 SAP의 관점에서 볼때 device block 들은 satellite 프로세서의 특성인 off-line I/O operation을 수행할 수 있다. 따라서 I/O device들의 특성에 맞는 interface unit의 설계에 의하여 독립된 I/O 채널의 구성이 가능하므로 Ⅲ. 1. 가. 항의 interface layer의 layer 2와 layer 3 (Device Dependent와 Device Independent)에 해당하는 device들을 모두 처리할 수 있는 기능을 갖고 있다.

또한 독립된 프로세서 단위의 device block은 I/O device와 해당 interface type (ISI, IPI, SCSI등)의 변경을 용이하게 하며 software layer의 설계에 의하여 Ⅱ. 2. 나항의 active/stand-by, continuation, paralle continuation 의 구성방식을 가능하게 하는 특성을 갖고 있다.

현재 TDX－1 시스팀의 I/O서브시스팀은〈그림 9〉에 보인바와 같이 system backup 용으로 CTIP를 active/stand by의 구조로, 과금과 통계자료용의 데이터는 MTIP, 데이터 채널 기능을 위한 DLIP, TTY와 CRT 및 Modem의제어를 위한 IOIP, 시스팀의 alarm source 처리를 하기 위한 SAIP로 구성되어 있다. 그러나 cartridge tape device는 serial access storage device의 형태이므로 파일 액세스에 상당한 시간이 걸리게 되어 시스팀의 복구시 요구되는 빠른 reloading의 기능이 부족한 상태이다.

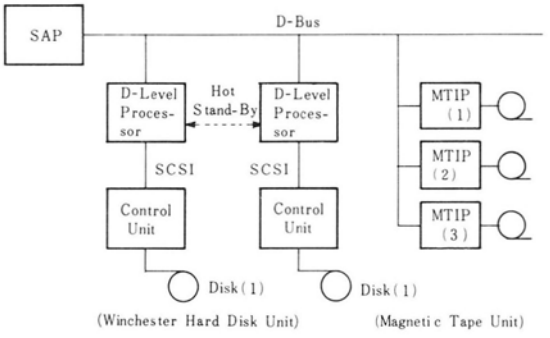
이에 대한 개선책으로는 msec. 단위의 액세스 시간을 갖는 Winchester hard disk의 direct access device를 사용하는 것이 바람직하다. 또한 I/O 채널의 인터페이스는 implementation의 용이성과 high-level command structure의 기능이 가능한 Ⅳ. 2. 절의 small computer system interface가 적합하다.

많은 양의 데이터를 계속 저장해야 하는 MTU (Magnetic Tape Unit)는 현재 2개의 MTU를 3개로 확장하고 처리방식은 Ⅱ. 2. 나. 절에서 설명한 continuation의 방식을 사용하는 것이 적합하다.



〈그림 9〉 I/O Processor의 구성 형태

앞에서 설명한 구성을 블럭 다이어그램 으로 나타내면 I/O storage 시스팀은 〈그림10〉과 같이 재구성 할 수 있다.



〈그림 10〉 I/O Processor중 Storage관련 Processor의 구성형태

V. 결 론

본고는 일반적인 I/O device의 처리 방식과 현재 ANSI에서 공인된 주변기기의 interface technique (ISI, IPI, SCSI)을 이용하여 교환기에서 요구하는 I/O 서브시스팀의 기능 등을 분류하고 이들을 실현할 수 있는 방법들을 설명하였다.

또한 TDX－1의 I/O프로세서 시스팀도 위의 방법들을 사용하여 분석한 후 현재의 CTU (Cartridge Tape Unit)를 Winchester hard disk로 대체해야 할 필요성과 implementation 시 D-level 프로세서 구조에 용이하게 적용할 수 있으며 high-level command structure의 사용이 가능한 SCSI architecture를 TDX－1 storage architecture에 적용하였다.

MTU (Magnetic Tape Unit)의 확장시에는 MTU 각각에 D-level 프로세서를 대응시키는 device-dependent 구조와 continuation의 처리 방식을 검토하였다.

〈參 考 文 献〉

1. "ANSC X3T9.2/82－2 Rev. 10" OEM Manual, Oct. 1983.
2. Madnick, Stuart E., John J. Donovan, Operating Systems.
3. Sideris, George, "Intelligent-Peripheral Interface", Electronics, Aug. 1982.
4. Allan, I. D., "Intelligent Peripheral Interface Updates Master-Slave Architecture", Electronics, Aug. 1982.
5. Henny T. Meyer, "Host Share Peripherals on Small-Computer Bus", Electronics, Sept. 1982.
6. Allan, Roger "Designer's Reference", Electronic Design, Dec. 1982.
7. Rubinson, Barry L., Rubinson, CMM. Riggle, "Disk-Storage Architecture Standardize Data Handling", Electronics, June. 1983.
8. Allan, I. D., "Controller Board Matches Intelligent Interface", Electronic Design, Apr. 1984.
9. Snively, Robert, "Multitasking Controller Speeds Throughput to Multiple Disks", Electronic Design, Jan. 1984.
10. I/O Software 개발, 한국전기 통신연구소, Dec 1983.
11. "IOS Functions in APZ 210", EN/LZT 101 137 RIA.