

HiPi-Bus의 최소 데이터 응답시간 (Minimum Data Response Time of HiPi-Bus)

奇安度 · 朴炳寬 · 尹龍鎬

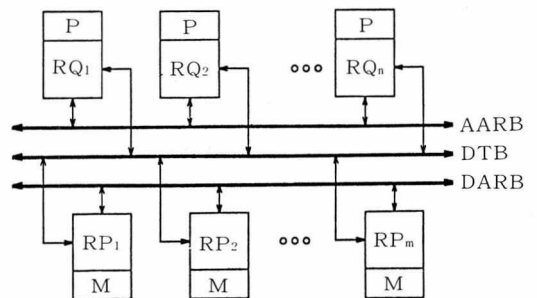
〈要 約〉

HiPi-Bus는 여러가지 전송형태를 지원하며 이들은 데이터 전송의 관점에서 읽기 전송과 쓰기 전송으로 구분할 수 있다. 본 논문에서는 이들 읽기 전송과 쓰기 전송에 소요되는 시간을 최소화하는 방법을 제안하였는데, 이때 쓰기 전송에 소요되는 시간은 400n sec이며, 읽기 전송에 소요되는 시간은 최소 400n sec이다.

I . ReQuester/ResPonder Model

HiPi-Bus(highly pipelined bus)의 데이터 전송 버스에서 데이터 전송은 데이터 전송 버스 요청기(ReQuester : RQ)와 데이터 전송 버스 응답기(ResPonder : RP) 사이에서 이루어진다. 일반적으로 RQ는 프로세서 보드에 위치하고, RP는 메모리 보드에 위치한다. 개념적으로 볼 때,

시스템 내의 데이터 전송은 프로세서와 메모리 사이에서 발생하고, RQ와 RP는 데이터 전송 통로인 데이터 전송 버스를 프로세서와 메모리



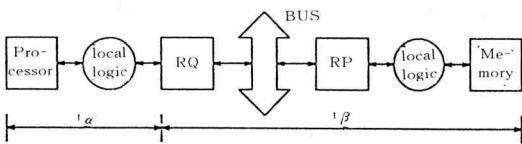
RQn : ReQuester
 RPm : ResPonder
 AARB : Address ARbitration Bus
 DARB : Data ARbitration Bus
 DTB : Data Transfer Bus
 P : Processor
 M : Memory

(그림 1) ReQuester/ResPonder Model

에 연결시켜 주는 역할을 담당한다. 따라서 데이터 전송 버스 전송 규격은 프로세서와 메모리는 모르는 상태에서 RQ와 RP에 의해 제어되고 수행된다.

데이터 전송 버스를 이용한 데이터 전송은 (그림1)과 같이 여러개의 RQ들과 여러개의 RP들이 데이터 전송 버스를 공유하여 사용하게 된다. 하나의 자원인 데이터 전송 버스가 여러 RQ와 여러 RP에 의해 공유되므로 이들 간의 중재를 하기 위한 중재 버스가 지원된다. RQ와 RP에 의해 수행되는 데이터 전송은 시스템 버스에서 공급되는 버스 클럭에 동기되어 이루어진다.

하나의 버스 동작을 생각할 때 데이터 경로는 (그림 2)와 같으며 프로세서에서 요청된 메모리 참조 요청은 버스를 통하여 RQ와 RP사이에서 이루어진다. 총 참조시간(total memory reference time)은 $t_{\alpha} + t_{\beta}$ 이다. t_{α} 는 프로세서에서 발생한 메모리 참조 요청이 RQ로 전달되고 RQ에서 수행한 버스 동작의 결과를 프로세서로 알리는 데 소요된 시간이며, 데이터 응답시간(data response time) t_{β} 는 프로세서로부터 전달받은 메모리 참조 요청을 RQ가 버스 규격에 맞게 버스 동작을 수행하여 메모리로부터 데이터를 참조(읽기 또는 쓰기)하는 데 소요된 시간이다.



(그림 2) Data Path

〈表 1〉 Transfer Types on HiPi-Bus

TT(3..0)*	Mnemonic	Name	Read/Write	Application
0×1	RFW	Read For Write	Read	Cache Coherency
0×2	INV	Invalidation	—	Cache Coherency
0×3	LCR	Interlock Read	Read	RMW cycle
0×4	NRD	Normal Read	Read	Normal
0×5	RFR	Read For Read	Read	Cache Coherency
0×8	NWR	Normal Write	Write	Normal
0×9	WRB	Write Back	Write	Cache Coherency
0×B	LCW	Interlock Write	Write	RMW cycle

이후 본 논문에서는 t_{β} 에 대해 상세히 살펴본다.

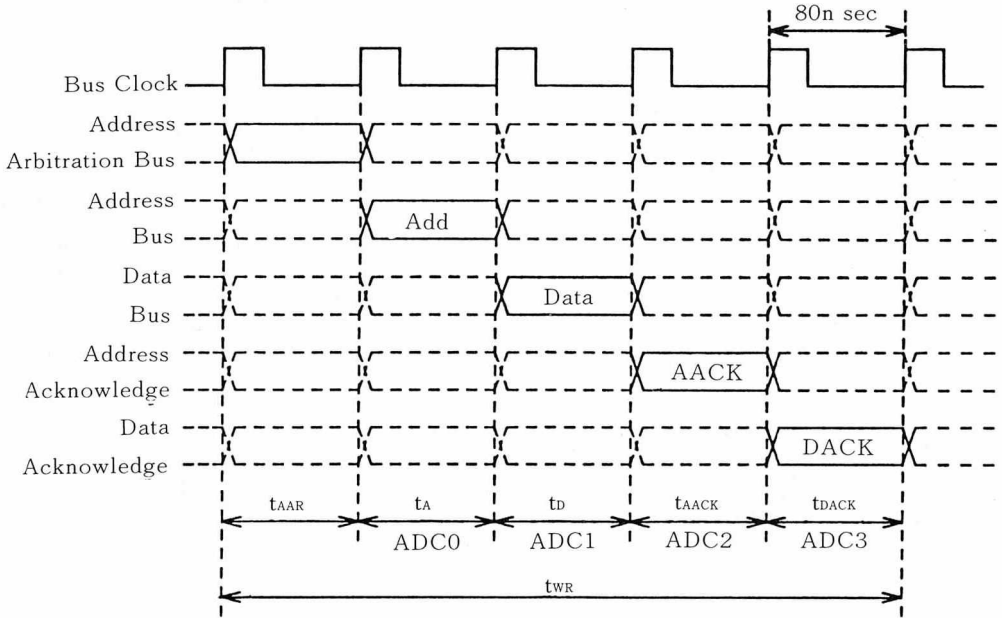
II . Data Transfer on HiPi-Bus

HiPi-Bus는 〈표 1〉과 같이 8가지의 전송형태를 지원하며, 이것은 수행하는 데이터 전송에 시스템에서 필요한 새로운 의미를 부여하여 버스의 기능을 고급화함으로써 시스템의 성능향상을 가능하게 한다. 그러나 데이터 전송은 버스 동작의 측면에서 읽기 전송, 쓰기 전송, 그리고 무효화 전송으로 구분될 수 있으며, 무효화 전송은 실제 데이터의 전송이 이루어 지는 것이 아니므로 여기에서는 읽기 전송과 쓰기 전송만을 고려한다.

1. Write Transfer

쓰기 전송은 RQ에서 RP로 데이터를 공급하는 것으로 어드레스 데이터 기본 주기만으로 구성된다. 어드레스 데이터 기본 주기는 RQ가 어드레스와 데이터 버스에 관련된 정보를 데이터 전송 버스에 구동하고, 선택된 RP는 상태 버스의 신호를 구동하여 RQ로부터 받은 정보에 대한 응답을 보내는 것으로 이러한 일련의 동작은 (그림 3)에 나타내었으며, 여기에서 어드레스 버스 중재 주기는 최초 1개 버스 사이클이 소요된다.

어드레스 데이터 기본 주기만으로 구성되는 쓰기 전송은 자체가 기본 주기로서 더 이상 소



(그림 3) Address Data Basic Cycle

요시간을 줄일 수 없다. 식 (1)에 쓰기 전송에 소요되는 시간을 식으로 나타내었다.

$$t_{WR} = t_{AAR} + t_A + t_D + t_{AACK} + t_{DACK} \dots\dots\dots (1)$$

특히, 식 (1)에서 각 시간은 <표 2>에 나타낸 값을 가지므로 쓰기 전송의 최소 소요시간은 식 (2)와 같다.

$$\begin{aligned} t_{WR} &= t_{AAR} + t_A + t_D + t_{AACK} + t_{DACK} \quad (\text{from 식(1)}) \\ &= t_{AAR} + 1 + 1 + 1 + 1 \\ &= t_{AAR} + 4 \end{aligned}$$

$$t_{WR} > 5 \dots\dots\dots (2)$$

따라서 식 (2)에 의해 쓰기 전송에 소요되는 최소시간은 5개 버스 클럭으로 400nsec(5×80n sec)이다. 그러나 어드레스 중재 주기를 제외하면 4개 버스 클럭으로 320nsec이며 이것은 쓰기 전송에 대한 RP의 순수한 응답시간이 된다.

2. Read Transfer

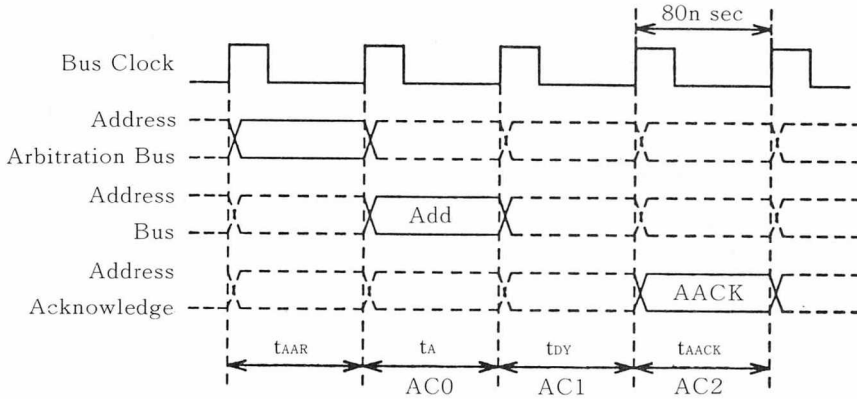
읽기 전송은 RQ에서 데이터를 RP로 요청하고 RP가 RQ로 해당 데이터를 공급하는 것으로,

어드레스 기본 주기와 데이터 기본 주기로 구성된다.

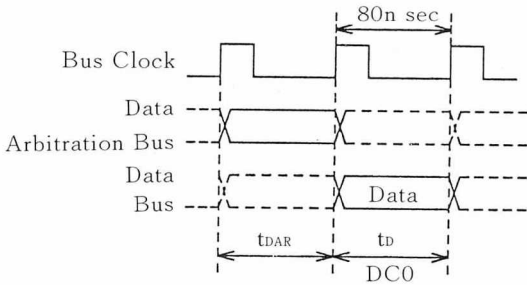
어드레스 기본 주기는 RQ가 RP에게 어드레스 버스에 관련된 정보를 전송하는 동작으로 RQ는 어드레스 버스 중재 주기를 수행한 후 버스 상의 모든 RP로 어드레스 버스의 정보를 보내고 RP들은 그 정보를 번역하고, 선택된 RP가 전달된 어드레스에 관한 정보에 대한 응답을 상태 버스를 통하여 보낸다. 이러한 일련의 동작은 (그림 4)와 같으며, 여기에서 어드레스 버스 중재 주기는 최소 1개 버스 사이클이 소요된다.

데이터 기본 주기는 RP가 RQ로 데이터를 공급하는 동작으로 RP는 데이터 버스 중재 주기를 수행한 후 데이터 버스에 관련된 정보를 데이터를 요청한 RQ로 보낸다. 이러한 일련의 동작은 (그림 5)에 나타내었다. 데이터 중재 주기는 최소 1개 버스 사이클이 소요된다.

어드레스 기본 주기와 데이터 기본 주기로 구성되는 읽기 전송중 가장 간단히 구성한 Simple Read Transfer Method(SRD)는 (그림 6)과 같이 두 기본 주기를 전혀 중첩없이 수행한 경우이며, 이때 버스 동작에 소요되는 시간은 식(3)



(그림 4) Address Basic Cycle



(그림 5) Data Basic Cycle

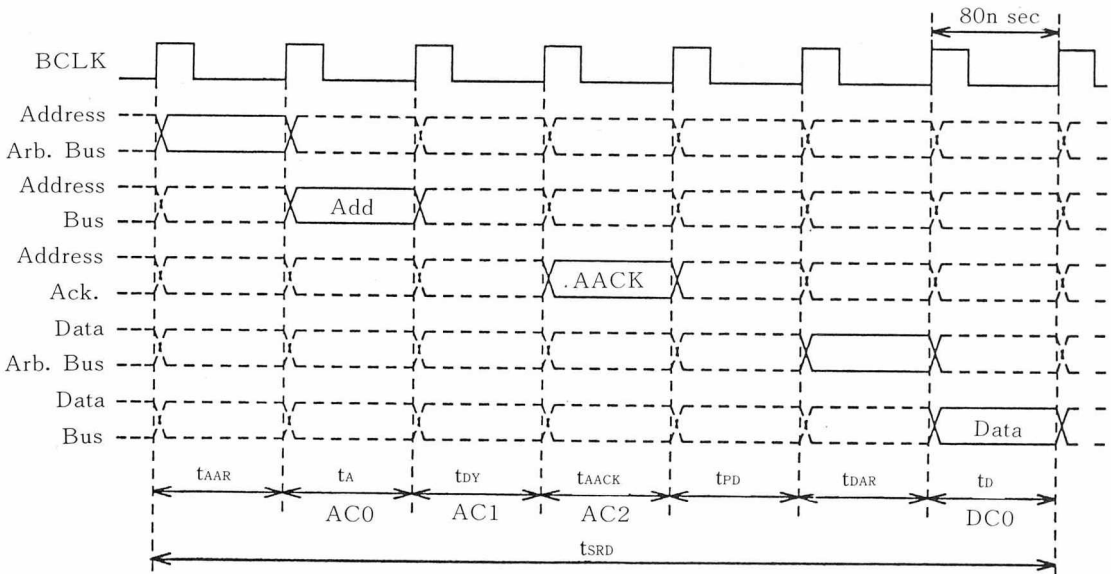
과 같다.

$$t_{SRD} = t_{AAR} + t_a + t_{DY} + t_{AACK} + t_{PD} + t_{DAR} + t_D \dots (3)$$

특히, 식 (3)에 나타난 각 시간은 <표 2>와 같은 값을 갖는다. 따라서 SRD에 소요되는 시간은 식 (3)과 <표 2>로부터 식 (4)와 같이 계산된다.

$$t_{SRD} = t_{AAR} + t_a + t_{DY} + t_{AACK} + t_{PD} + t_{DAR} + t_D$$

(from 식(3))



(그림 6) Simple Read Transfer Method

$$= t_{AAR} + 1 + 1 + 1 + t_{PD} + t_{DAR} + 1$$

$$= 4 + t_{AAR} + t_{DAR} + t_{PA}$$

$$t_{SRD} > 7 \dots\dots\dots (4)$$

그러나 SRD에서 중첩하여 진행 가능한 것이 있으며, 이러한 중첩 진행을 구현할 때 소요시간을 줄일 수 있다. 어드레스 기본 주기와 데이터 기본 주기 내부는 더 이상 중첩이 안되는 것이지만 두 기본 주기를 중첩할 수 있다.

즉, (그림 6)에서 어드레스가 전달되는 AC0 이후에는 RP에서 내부 메모리 내용을 준비할 수 있으므로 데이터를 준비하는 동작과 AC0 이후의 어드레스 기본 주기는 중첩 가능하며, 또한 데이터를 준비하는 동작이 미리 예측 가능한 시간내에 이루어진다면 이것과 데이터 버스 중재 주기 역시 중첩 가능하다. 이러한 것을 고려한 Minimum Read Transfer Method(MRD)는 (그림 7)과 같이 된다. 그러나 어드레스 응답 주기 AC2와 데이터 주기 DC0 사이의 시간 t_δ 는 0보다 작을 수는 없다. 즉 데이터 주기가 어드레스 응답 주기보다 선행 또는 동시에 이루어

질 수는 없다는 것이다. 또 데이터 준비 시간 t_{PD} 는 데이터 버스 중재 시간 t_{DAR} 에 버스 클럭 1 개를 더한 값 $t_{DAR} + 1$ 보다 작거나 같아야 된다. 이것은 데이터가 준비되지 않은 상태에 데이터 중재가 완료되지 않음을 보장하는 것이다. 이 두 조건은 식 (5)와 식 (6)으로 표현된다.

$$t_\delta \geq 0 \dots\dots\dots (5)$$

$$t_{PD} \leq t_{DAR} + 1 \dots\dots\dots (6)$$

따라서 MRD에 소요되는 시간은 식 (7)과 같다.

$$t_{MRD} = t_{AAR} + t_A + t_{DY} + t_{AACK} + t_D \dots\dots\dots (7)$$

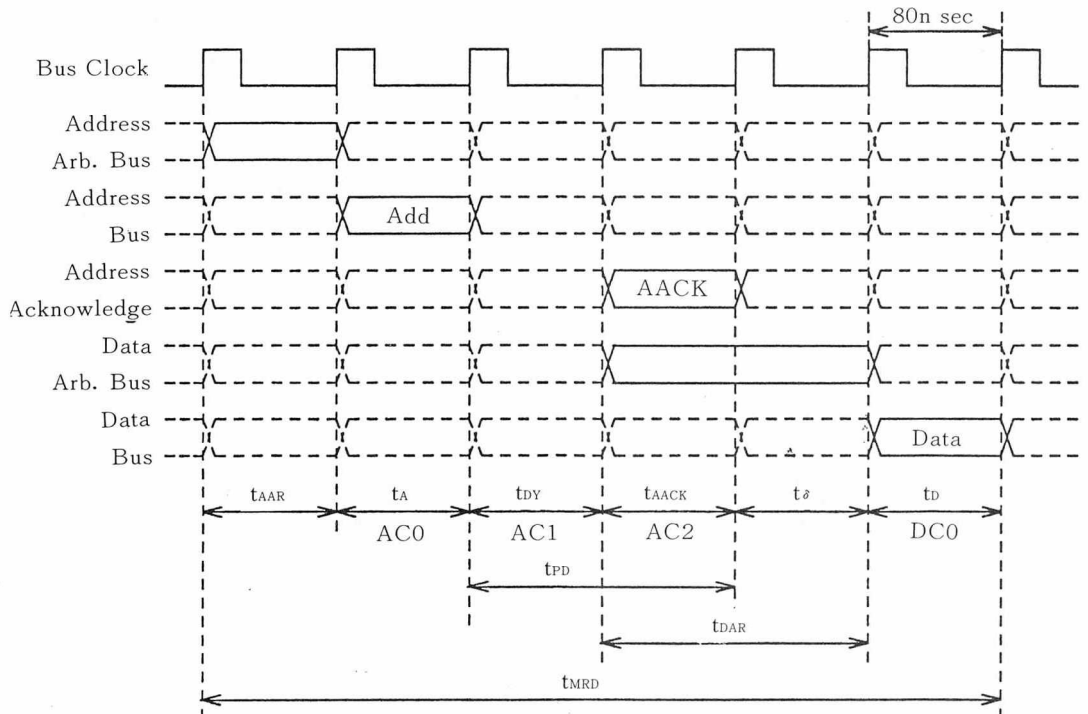
식 (7)에 최소값만을 대입하면 식 (8)과 같이 MRD의 최소 소요시간이 계산된다.

$$t_{MRD} = t_{AAR} + t_A + t_{DY} + t_{AACK} + t_\delta + t_D \text{ (from 식(7))}$$

$$= t_{AAR} + 1 + 1 + 1 + t_\delta + 1$$

$$t_{MRD} \geq 5 \dots\dots\dots (8)$$

식 (8)을 만족시키기 위한 조건인 식 (6)에서 RP에 있는 메모리의 응답시간이 계산될 수 있



(그림 7) Minimum Read Transfer Method

으며 이것은 식 (9)와 같다.

$$\begin{aligned}
 t_{PD} &\leq t_{DAR} + 1 \quad (\text{from 식 (6)}) \\
 t_{DAR} &\geq 1 \quad (\text{from <표 2>)} \\
 t_{PD} &\leq 1 + 1 \\
 t_{PD} &\geq 2 \dots\dots\dots (9)
 \end{aligned}$$

식 (9)에서 알 수 있듯이 MRD를 구현하는 데 필요한 RP내 메모리의 응답시간은 버스 클럭 2개(2×80nsec=160nsec)보다 빨라야 한다. 따라서 식 (8)에 의해 읽기 전송에 소요되는 최소 시간은 5개 버스 클럭으로 400nsec(5×80nsec)이다. 그러나 어드레스 중재 주기를 제외하면 4개 버스 클럭으로 320n sec이며 이것은 읽기에 대한 RP의 순수한 응답시간이 된다.

<표 2> Minimum and Maximum Values

	meaning	min	max
t _{AAR}	Address Bus Arbitration	1	‡
t _A	drive Address	1	1
t _{DY}	Dummy	1	1
t _{AACK}	Address Acknowledge	1	1
t _{PD}	Prepare Data	1*	—
t _{DAR}	Data Bus Arbitration	1	‡
t _D	drive Data	1	1
t _{DACK}	Data Acknowledge	1	1
t _‡	time interval between AACK and Data on Read Transfer	0	—

(unit : Bus Cycle, 80nsec)

* : RP의 메모리 응답시간이 80nsec 이내인 경우
 ‡ : 중재에 소요되는 시간은 상태에 따라 변한다.

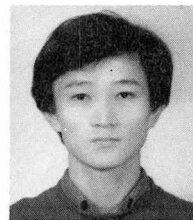
III . Conclusions

HiPi-Bus는 여러가지 전송형태를 지원하며 이들은 데이터 전송의 관점에서 읽기 전송과 쓰기 전송으로 구분할 수 있다. 읽기 전송은 어드레스 데이터 기본 주기로 수행되며, 쓰기 전송은 어드레스 기본 주기와 데이터 기본 주기로 수행된다. 쓰기 전송은 그 자체가 기본 주기이므로 소요시간을 줄일 수 없으며, 최소 소요 시간은 5개 버스 클럭으로 400nsec이다. 읽기 전

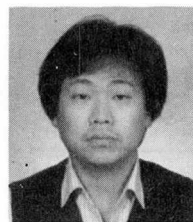
송은 두 기본 주기를 수행하는 RP의 동작과 RP와 메모리간의 동작을 최대로 중첩하여 수행할 때 소요시간을 줄일 수 있다. 즉, RP를 어드레스 기본 주기에 관련된 동작, 내부 메모리를 참조하는 동작, 그리고 데이터 기본 주기에 관련된 동작을 독립적으로 수행 가능하도록 구현하여 이들을 최대한 중첩하여 동작되도록 한다. 이때 최소 소요시간은 5개 버스 클럭으로 400nsec이다.

<參 考 文 獻>

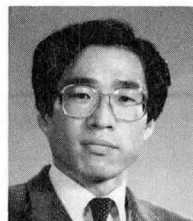
1. 컴퓨터구조연구실, “타이컴 시스템 버스 사용자 매뉴얼”, TD88-6120-133.B, 행정전산망 주전산기 개발본부, 한국전자통신연구소, 1990.



奇安度 (Ki, An-Do)
 1982. 3—1986. 2 : 한양대학교 전자공학과 학사
 1986. 3—1988. 2 : 한국과학기술원 전기 및 전자공학과 석사
 1988. 3—1991 현재 : 한국전자통신연구소 컴퓨터구조연구실 연구원



朴炳寬 (Park, Byung Kwan)
 1978. 3—1982. 2 : 한양대학교 전자공학과 학사
 1985. 3—1990. 8 : 한국과학기술원 전산학과 석사
 1982—1991 현재 : 한국전자통신연구소 컴퓨터구조연구실 선임연구원



尹龍鎬 (Yoon, Yong Ho)
 1970—1975 : 한양대학교 전자공학과 학사
 1978—1981 : 연세대학교 대학원
 1985—1989 : 한양대학교 대학원 박사과정 수료
 1975—1978 : 한국과학기술연구소 컴퓨터국산화연구실

1978—1991 현재 : 한국전자통신연구소 컴퓨터구조연구실 실장, 책임연구원