AESOP Embedded Forum
Sejong Lee (http://www.aesop-embedded.org)

개 요

이 문서는 S3C6400 데이터시트에서
제 27장 SD/MMC Host Controller를 번역한 자료입니다.

제 27장. SD/MMC 호스트 콘트롤러

이번 장은 S3C6400 RISC 마이크로프로세서에서 제공하는
SD/SDIO/MMC/CE-ATA 호스트 콘트롤러와
이와 관련된 레지스터를 설명하고 있습니다.

27.1 개관

SD/MMC 호스트 콘트롤러는 Secure Digital Card 와 MultiMediaCard 를 위해 연합된 호스트이다. 이 호스트는 SD 협회(SDA)의 표준 스펙과 호환된다.

사용자는 SD 카드나 MMC 카드로 시스템을 연결할 수 있다. 이 호스트의 성능은 매우 강력하며 사용자는 50MHz 의 클럭 속도를 얻을 수 있고 동시에 8 비트 데이터 핀을 엑세스할 수 있다.

27.2 특징

High-Speed MMC 콘트롤러 지원

- SD Standard Host Specification (ver 1.0) 호환
- SD Memory Card Specification (ver 2.0) / HSMMC Specification (4.0) 호환
- SDIO Card Specification (ver 1.0) 호환
- 512 바이트 FIFO for data Tx/Rx
- CPU 인터페이스와 DMA 데이터전송 모드
- 1 비트/4 비트/8 비트 모드전환 지원
- Auto CMD12 지원
- Suspend/Resume 지원
- Read Wait 동작 지원
- Card Interrupt 지원
- CE-ATA 모드 지원

27.3 BLOCK DIAGRAM

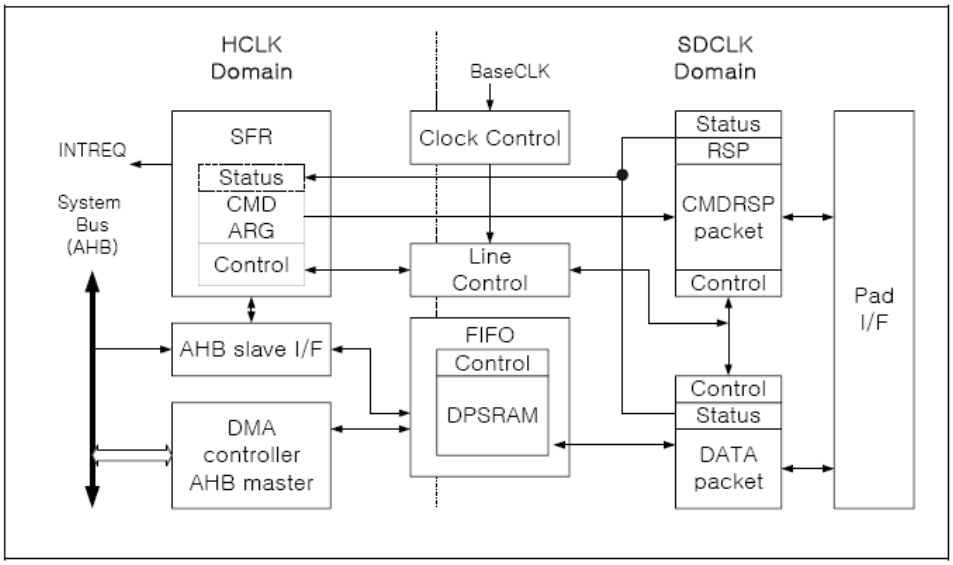

Figure 27-1. HSMMC block diagram

27.4 Sequence (절차)

이번 단락은 여러개의 하위 시퀀스로 나뉘어지는 basic sequence flow(기본 시퀀스 흐름)을 정의한다. "Wait for interrupts"는 플로우 차트에서 사용된다. 이것은 호스트 드라이버가 특정된 인터럽트가 있을 때까지 기다린다는 의미이다. 만약 인터럽트가 이미 떳다면 플로우 차트에서 다음 단계를 따른다. timeout 체크는 인터럽트가 미발생되었음을 감지하는 것이 요구된다. 이것은 플로우 차트에 설명되어 있지 않다.

27.4.1 SD 카드 감지 절차

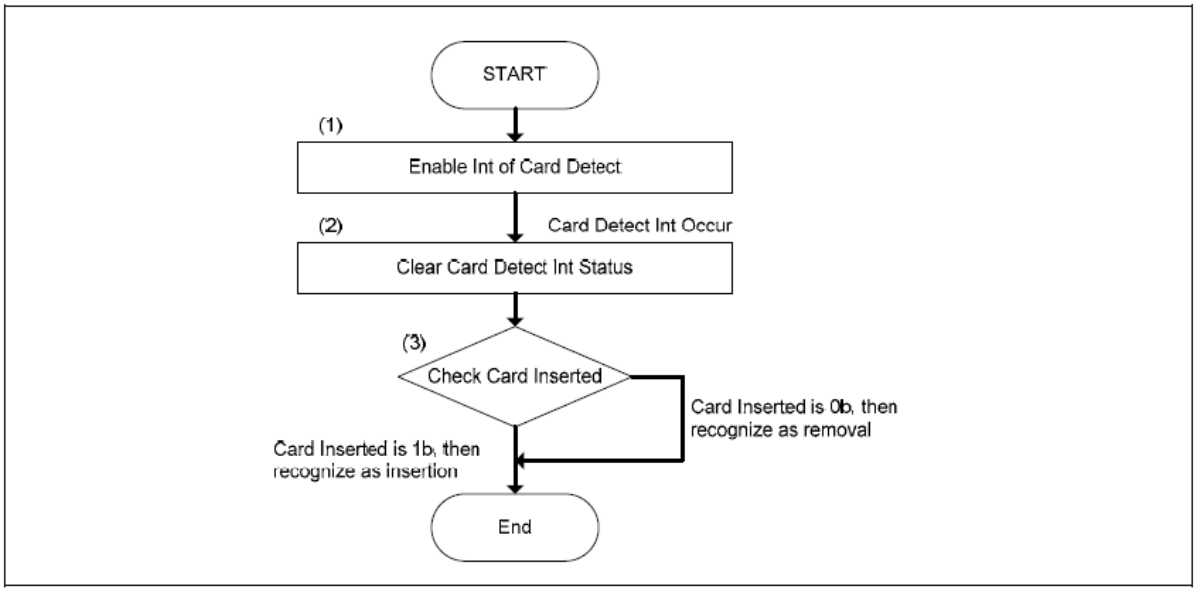

Figure 27-2. SD Card Detect Sequence

SD 카드 감지에 대한 플로우 차트는 Figure 27-2 에 나타내었다. 각각의 단계는 다음과 같이 실행된다.

(1) 카드 감지를 위한 인터럽트를 활성화하기 위해 다음의 비트에 1 을 쓴다.

Card Insertion Status Enable(ENSTACARDNS) in the Normal Interrupt Status Enable register
Card Insertion Signal Enable(ENSIGCARDNS) in the Normal Interrupt Signal Enable register
Card Removal Status Enable(ENSTACARDREM) in the Normal Interrupt Status Enable register
Card Removal Signal Enable(ENSIGCARDREM) in the Normal Interrupt Signal Enable register

(2) 호스트 드라이버가 카드 삽입 또는 제거를 감지했을 때, 호스트 드라이버는 interrupt status 를 clear 한다. 만약 카드 삽입 인터럽트(STACARDINS)가 발생했다면 Normal Interrupt Status register 의 Card Insertion 에 1 을 쓴다. 만약 카드 제거 인터럽트(STACARDREM)가 발생했다면 Normal Interrupt Status register 의 Card Removal 에 1 을 쓴다.

(3) Present State register 의 Card Inserted 를 체크한다. Card Inserted(INSCARD)가 1 일 경우, 호스트 드라이버는 전원과 클럭을 SD 카드로 공급할 수 있다. Card Inserted 가 0 일 경우, 호스트 드라이버의 다른 실행 프로세스들은 즉시 종료될 것이다.

27.4.2 SD 클럭 공급 절차

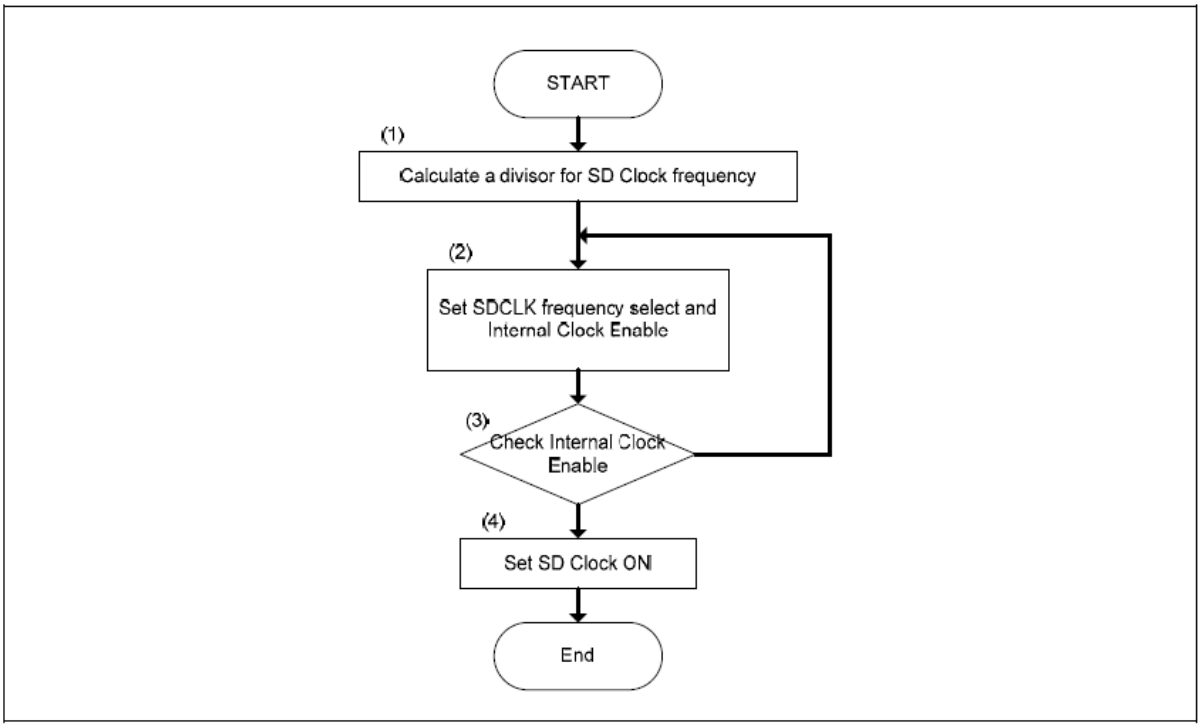

Figure 27-3. SD Clock Supply Sequence

SD 카드로 SD 클럭을 공급하기 위한 절차는 Figure 27-3 에 설명되어 있다. 클럭은 다음의 동작중 하나가 취해지기전에 공급되어야 한다.

a) Issuing a SD command (SD 커맨드 실행)

b) 4 비트 모드의 SD 카드로부터 인터럽트 감지

(1) SD 클럭 주파수를 결정하기 위해 divisor(제수, 나눗수) 계산 (for SD Clock by reading Base Clock Frequency)
clock control register 참고

(2) 단계(1)의 계산된 결과에 따라 Clock Control register 의 Internal Clock Enable(ENINTCLK)과 SDCLK Frequency Select 설정

(3) Clock Control register 의 Internal Clock Stable(STBLINTCLK) 체크. Clock Stable 값이 1 일때까지 이 단계를 반복한다.

(4) Clock Control register 의 SD Clock Enable(ENSDCLK)을 1 로 설정한다. 그러면 호스트 콘트롤러는 SD 클럭을 공급하기 시작한다.

27.4.3 SD 클럭 정지 절차

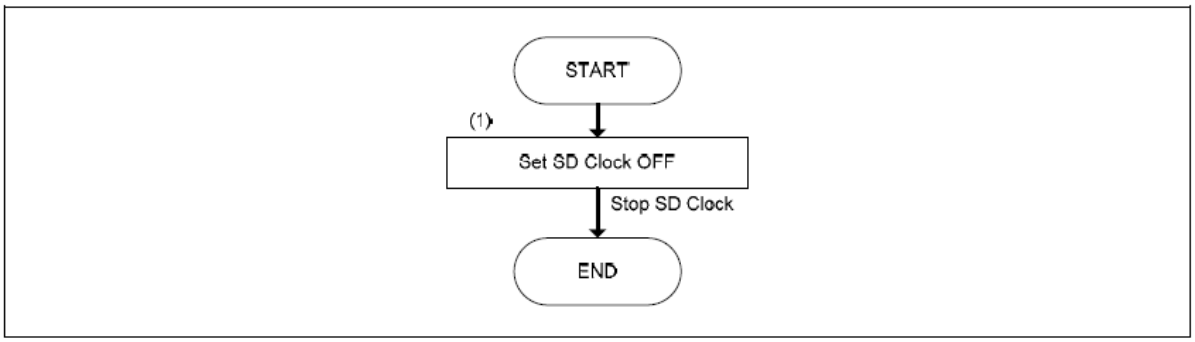

Figure 27-4. SD Clock Stop Sequence

SD 클럭 정지에 대한 플로우차트는 Figure 27-4 에 나타내었다. 호스트 드라이버는 SD 클럭을 멈추지 않는다, SD transaction(데이터처리)가 SD 버스상에서 발생할때에, 즉, Present State register 의 Command Inhibit(DAT) 또는 Command Inhibit(CMD)가 1 로 설정되었을 때를 말한다.
(1) Clock Control register 의 SD Clock Enable(ENSDCLK)를 0 으로 설정한다. 그러면 호스트 콘트롤러는 SD 클럭 공급을 정지한다.

27.4.4 SD 클럭 주파수 변경 절차

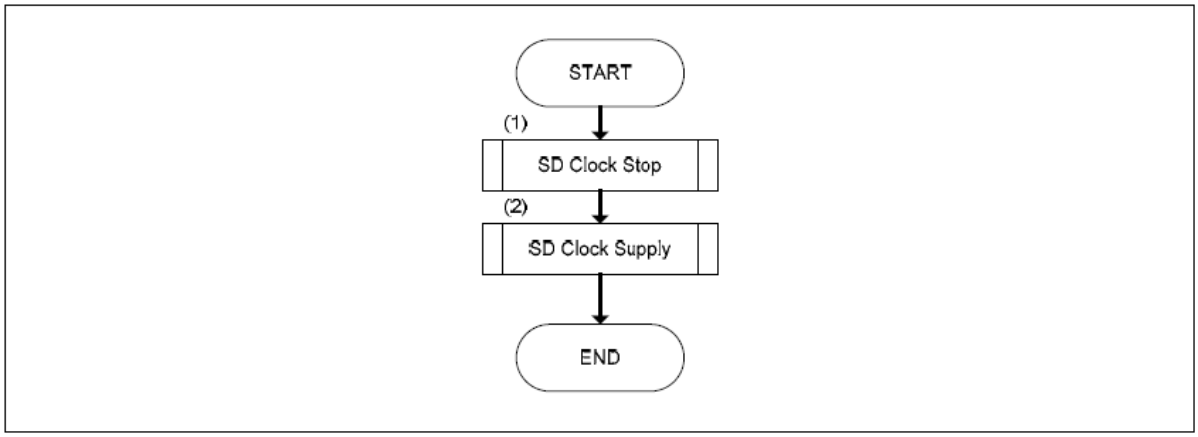

Figure 27-5. SD Clock Change Sequence

SD 클럭 주파수 변경 절차를 Figure 27-5 에 나타내었다. SD 클럭이 still off 일 때, (1)단계는 생략된다.

(1) SD 클럭 정지 절차를 수행한다. 27.4.2 참고

(2) SD 클럭 공급 절차를 수행한다. 27.4.3 참고

27.4.5 SD 버스 전력제어 절차

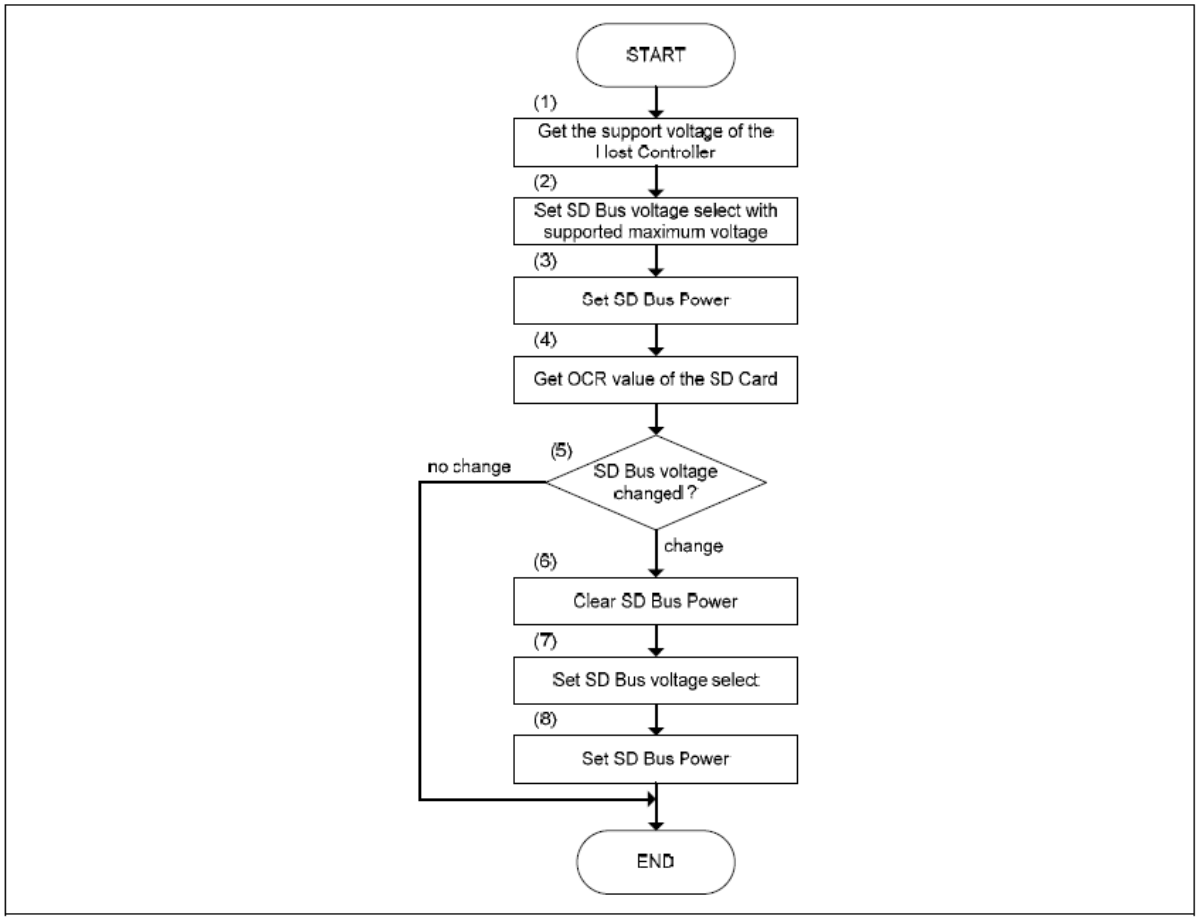

Figure 27-6. SD Bus Power Control Sequence

SD 카드 전력을 제어하기 위한 절차는 Figure 27-6 에 설명되어 있고 단계들은 아래에 나타내었다.

(1) Capabilities register 를 읽어서 호스트 콘트롤러의 지원가능한 전압을 가져온다.

(2) 최고 전압을 가지고 있는 외부 파워 레귤레이터(옵션)의 SD Bus Voltage Select 를 설정한다. 이것은 호스트 콘트롤러가 지원한다.

(3) Power Control register 의 SD Bus Power(PWRON)을 1 로 설정한다.

(4) SD 카드 내부의 모든 기능에 대한 OCR 값을 가져온다.

(5) SD 버스의 전압이 변경되어야 하는지, 아닌지를 판단한다. 만약 SD 버스의 전압이 변경되어야 한다면, 단계(6)으로 가라. 만약 SD 버스 전압이 변경되지 않는다면 go to 'End'.

(6) Power Control register 의 SD Bus Power 를 0 으로 설정한다.(이 비트를 clear 하기 위해) 카드가 올바르게 감지되기 위해서는 0 볼트에서의 전압상승이 요구된다. 호스트 드라이버는 SD Bus Power 를 clear 할 것이다, SD Bus Voltage Select 를 설정해서 전압이 변경되기 전에.

(7) Power Control register 의 SD Bus Voltage Select(SELPWRLVL)을 설정한다.
(8) Power Control register 의 SD Bus Power 를 1 로 설정한다.

NOTE: 단계(2)와 단계(3)은 동시에 실행될 수 있다. 또한 단계(7)과 단계(8)도 동시에 실행될 수 있다.

27.4.6 Change Bus Width Sequence (버스폭 변경 절차)

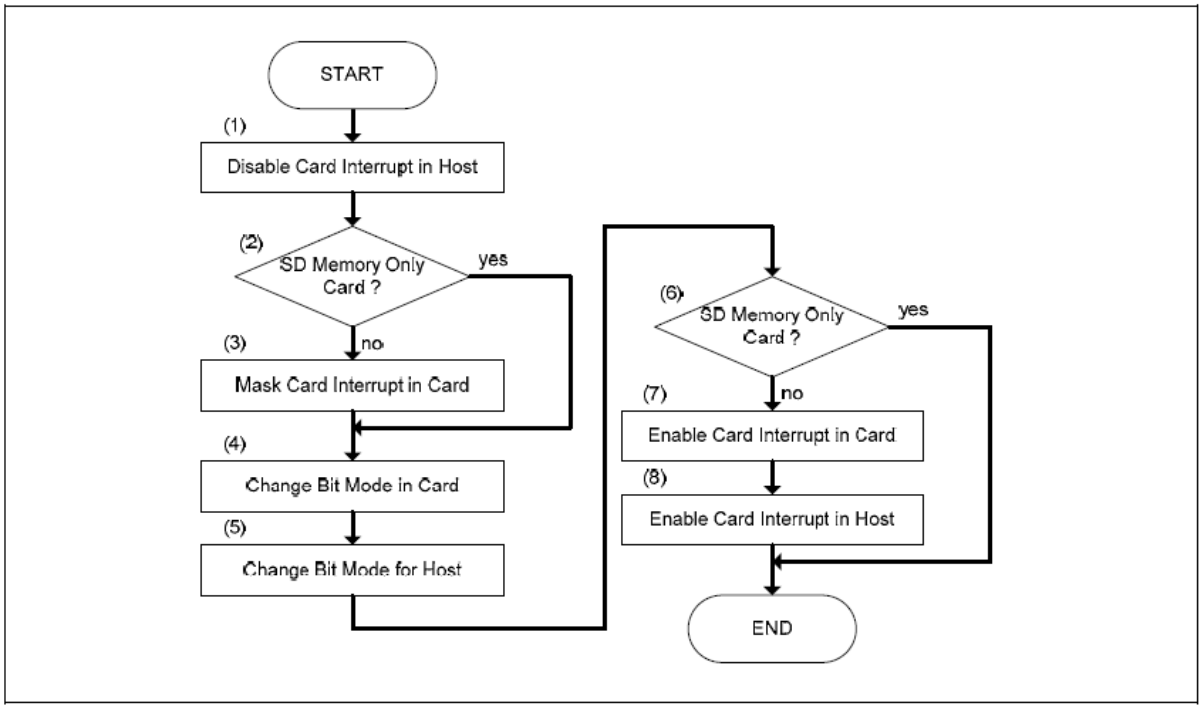

Figure 27-7. Change Bus Width Sequence

SD 버스의 비트변경 모드에 대한 절차는 Figure 27-7 에서 보여주고 있다.

(1) Normal Interrupt Status Enable register 의 Card Interrupt Status Enable(STACARDINT)를 0 으로 설정한다, 올바르지 않은 인터럽트를 masking(차폐)하기 위해. 이것은 버스폭이 변경될 동안 일어날 수 있다.

(2) SD memory card 의 경우에는 go to step(4). 다른 카드의 경우에는 go to step(3).

(3) SDIO 나 SD combo card 의 OCR 에 있는 IENM 을 0 으로 설정한다, CMD52 를 사용해서.

(4) SD 카드를 위해 비트모드를 변경한다. SD memory card 의 버스폭은 ACMD6(Set bus width)에 의해 변경하고 SDIO 카드의 버스폭은 CCCR 에 있는 Bus Interface Control register 의 Bus Width 를 설정함으로서 변경한다.

(5) 사용자가 4 비트 모드로 변경하길 원하는 경우에는, Host Control register 에 있는 Data Transfer Width(WIDE4)를 1 로 설정한다. 다른 경우에는(1 비트 모드), 이 비트를 0 으로 설정한다.

(6) SD memory card 의 경우, go to the 'End'. 다른 카드의 경우, go to step(7).

(7) SDIO 나 SD combo card 에 있는 CCCR 의 IENM 을 1 로 설정한다, CMD52 를 사용해서.

(8) Normal Interrupt Status Enable register 의 Card Interrupt Status Enable 을 1 로 설정한다.

27.4.7 Timeout Setting for DAT Line (DAT 라인을 위한 시간제한 설정)

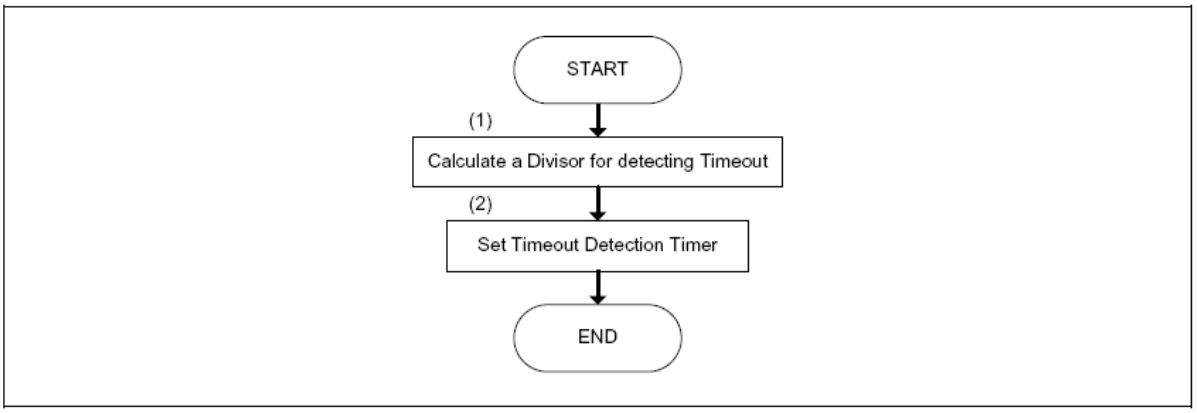

Figure 27-8. Timeout Setting Sequence

DAT line 의 timeout 에러를 감지하기 위해, 호스트 드라이버는 어떤 SD 데이터처리 이전에 다음의 두 가지 단계를 수행한다.

(1) timeout 을 감지하기 위해 divisor(제수, 나눗수)를 계산한다, Timeout Control Register 를 참고하자.
(2) Timeout Control register 의 Data Timeout Counter(TIMEOUTCON)을 설정한다, 위의 단계(1)의 값에 따라서.

27.4.8 SD Transaction Generation (SD 데이터처리 생성)

이번 단락은 여러가지 SD 트랜잭션(데이터처리)을 생성하는 것과 제어하는 것에대한 절차를 설명하고 있다. SD 트랜잭션은 3 가지의 경우로 나눌 수 있다.

(1) DAT 라인을 사용하지 않는 트랜잭션
(2) DAT 라인을 사용하는 트랜잭션 (only for the busy signal.)
(3) DAT 라인을 사용하는 트랜잭션 (for transferring data.)

이 스펙에서 첫 번째와 두 번째 경우의 트랜잭션은 “Transaction Control without Data Transfer using DAT Line"으로 분류되고, 세 번째 경우의 트랜잭션은 ”Transaction Control with Data Transfer using DAT Line"으로 분류된다.

SD Command 에 대한 자세한 사항은 아래의 스펙을 참고하기 바란다.

* SD Memory Card Specification Part1
PHYSICAL LAYER SPECIFICATION Version 1.01
* SD Card Specification PART E1
Secure Digital Input/Output(SDIO) Specification Version 1.00

27.4.9 SD Command 수행 절차

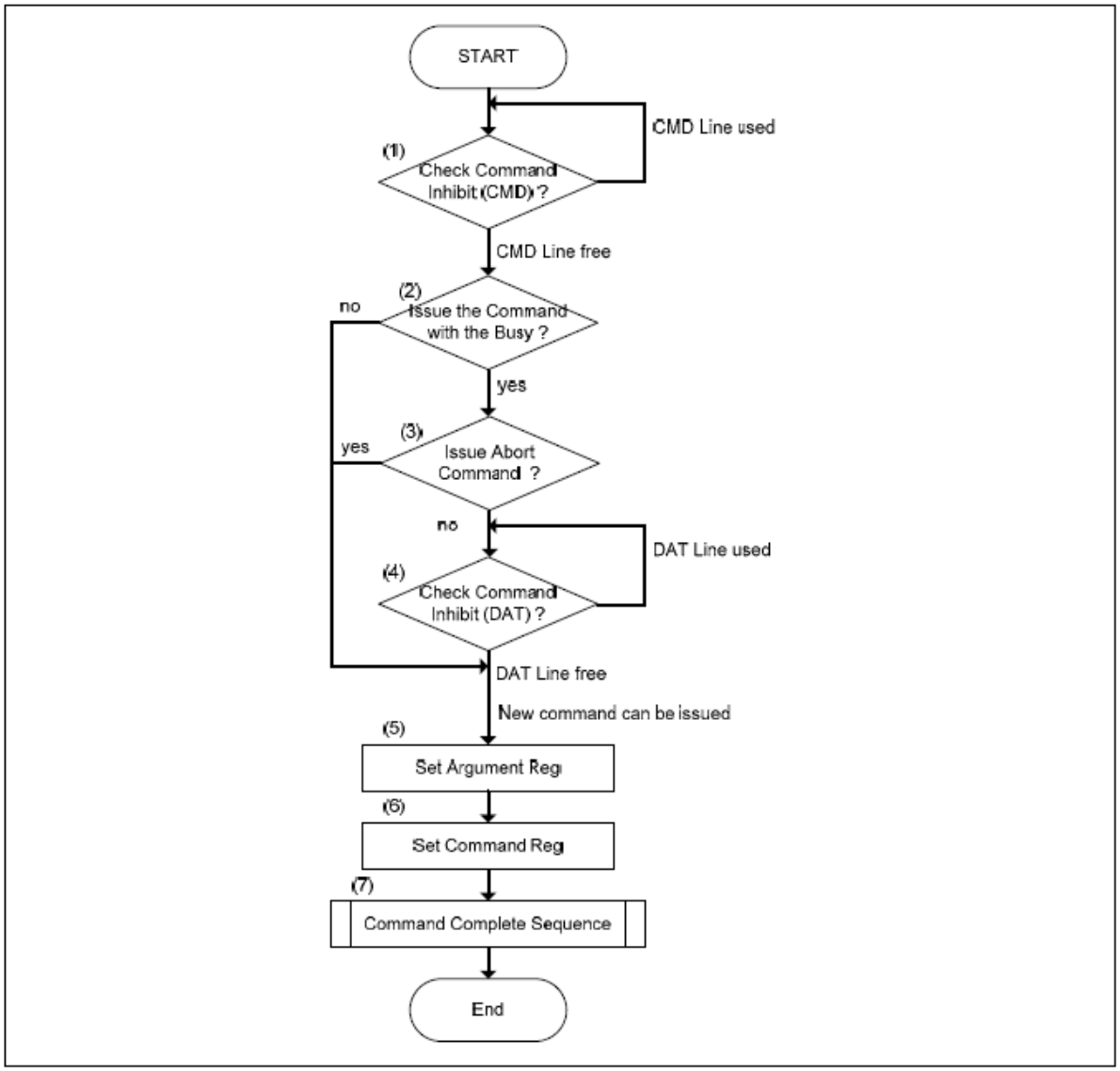

Figure 27-9. Timeout Setting Sequence

Timeout 설정을 위해 다음의 단계를 취한다.

(1) Present Status register 의 Command Inhibit(CMD)를 체크한다. Command Inhibit(CMD)가 0 일때까지 이 단계를 반복한다. Command Inhibit(CMD)가 1 일때, 호스트 드라이버는 SD Command 를 실행하지 않는다.
(2) 만약 호스트 드라이버가 busy signal 로 SD Command 를 실행한다면, go to step(3). 만약 busy signal 이 없다면, go to step(5).
(3) 만약 호스트 드라이버가 abort command 를 실행한다면, go to step(5). no abort command 의 경우에는, go to step(4).
(4) Present State register 의 Command Inhibit(DAT)를 체크한다. Command Inhibit(DAT)가 0 일때까지 이 단계를 반복한다.
(5) Argument register 의 실행된 Command 에 따라서 값을 설정한다.
(6) Command register 의 실행된 Command 에 따라서 값을 설정한다.

Note: Command register 에서 upper byte(상위바이트)에 writing 하는 것은 SD command 가 실행되는 것의 원인이 된다.
(7) Command Complete Sequence 를 수행한다.

27.4.10 Command Complete Sequence (Command 완료 절차)

SD Command 완료에 대한 절차는 Figure 27-10 에 나타나 있다. 이 절차가 수행되는 동안에 에러(Command Index/End bit/CRC/Timeout Error)가 발생할 가능성이 있다.

(1) Command Complete Interrupt 를 기다린다. 만약 이 인터럽트가 발생하면, go to step(2)
(2) Normal Interrupt Status register 의 Command Complete(STACMDCMPLT)에 1 을 쓴다, 이 비트를 clear 하기 위해.
(3) Response register 를 읽고 실행된 command 에 따른 필요한 정보를 가져온다.

(4) command 가 Trnasfer Complete Interrupt 를 사용했는지 아닌지를 판단한다. 만약 Transfer Complete 를 사용했다면, step(5)로 계속해서 수행한다. 그렇지 않다면, go to step(7).

(5) Transfer Complete Interrupt 를 기다린다. 만약 이 인터럽트가 발생하면, go to step(6).

(6) Normal Interrupt Status register 의 Transfer Complete(STATRANCMPLT)에 1 을 쓴다, 이 비트를 clear 하기 위해.

(7) Response Data 에서 에러가 있는지 체크한다. 만약 에러가 없다면 step(8)로 계속 진행한다. 만약 에러가 있다면, go to step(9).

(8) "No Error.“ 상태를 반환한다.

(9) “Response Contents Error." 상태를 반환한다.

NOTES:

1. Transfer Complete interrupt 를 기다리는 동안, 호스트 드라이버는 busy signal 을 사용하지 않고 단지 command 만 실행한다.

2. 호스트 드라이버는 Transfer Complete 를 감시하면서 Auto CMD12(Stop Command)가 완료되었는지 판단한다.

3. memory multiple blocks read command(CMD18)을 사용해서 비보호된 지역의 마지막 블록을 읽었을 때, OUT_OF_RANGE 에러가 발생할 수 있다, 비록 그 절차가 올바르더라도. 호스트 드라이버는 이것을 무시한다. 이 에러는 Auto CMD12 response(응답) 또는 다음 memory command 의 응답에 나타난다.

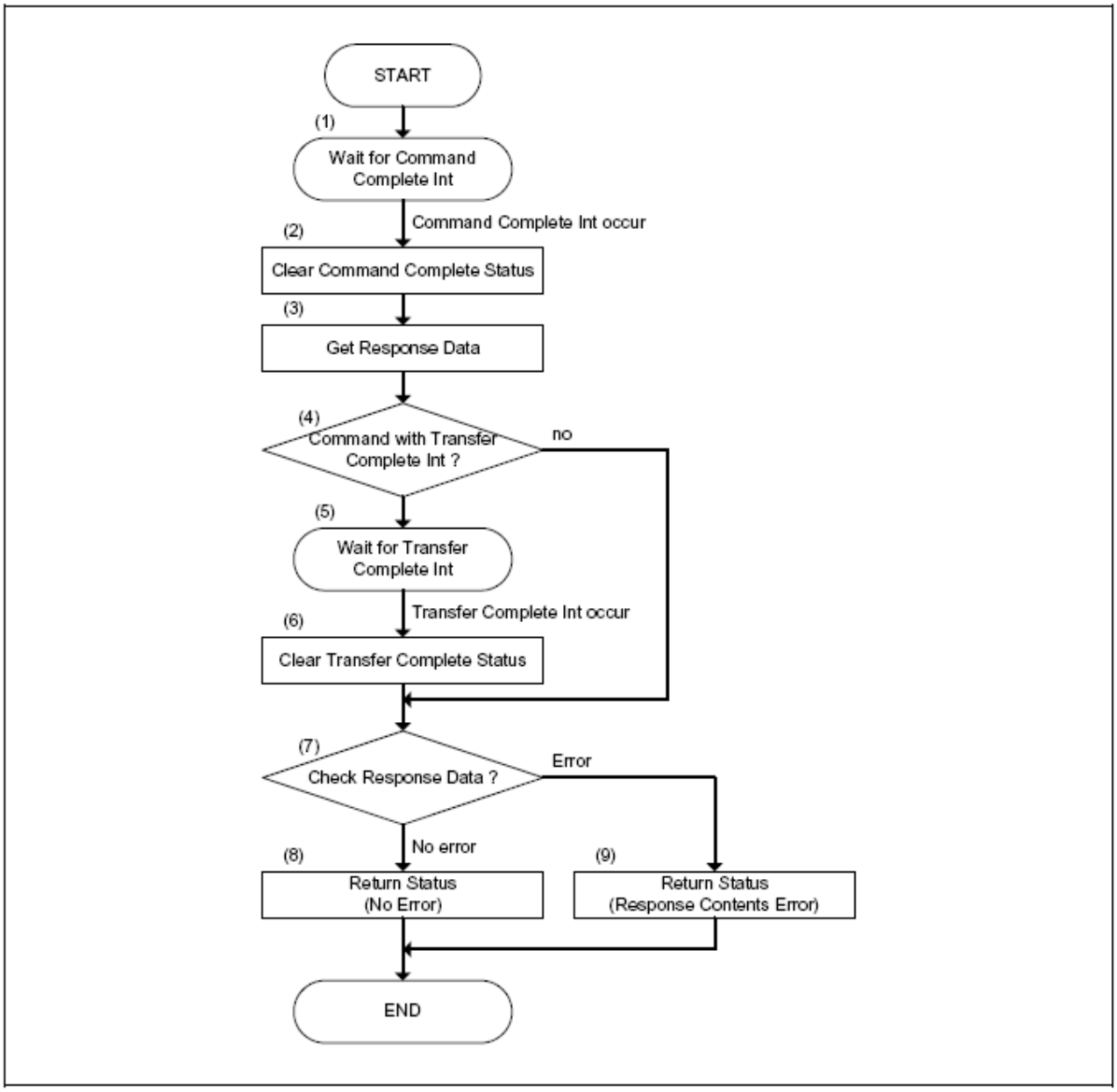

Figure 27-10. Command Complete Sequence

27.4.11 DAT 라인을 사용한 데이터 전송에서 트랜잭션(데이터처리) 제어

DMA(옵션)가 사용되는지 아닌지에 따라, 두 개의 수행 방법이 있다. DMA 를 사용하지 않는 절차는 Figure 27-11 에서 보여주고 있고 DMA 를 사용하는 절차는 Figure 27-12 에 나타나 있다.

추가로, SD 전송에 대한 절차는 기본적으로 얼마만큼의 블록수가 규정되어 있는지에 따라서 분류된다. 3 가지 종류의 분류는 다음과 같다.

1) Single Block 전송
블록개수는 전송되기 전에 호스트 콘트롤러가 지정한다. 지정된 블록개수는 언제나 하나이다.

2) Multiple Block 전송
블록개수는 전송되기 전에 호스트 콘트롤러가 지정한다. 지정된 블록개수는 하나 또는 그 이상일 것이다.

3) Infinite Block 전송
블록개수는 전송되기 전에 호스트 콘트롤러가 지정하지 않는다. 이 전송은 트랜잭션 중단이 실행되기전까지 계속된다. 이 트랜잭션 중단은 SD memory card 의 경우 CMD12(Stop Command)에 의해 SDIO 카드의 경우 CMD52(IO_RW_DIRECT)에 의해 수행된다.

27.4.12 NOT USING DMA (DMA 미사용)

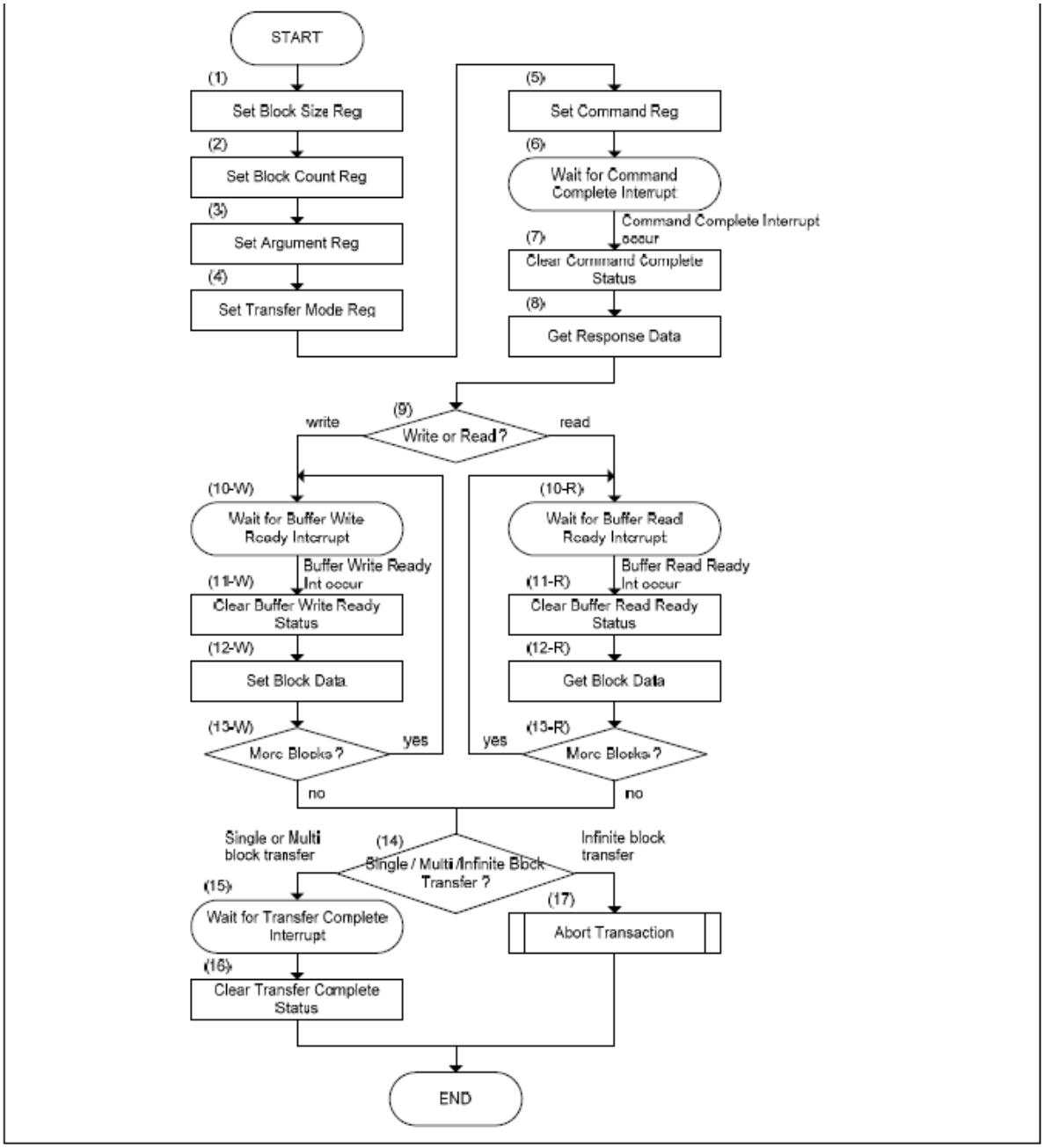

Figure 27-11. Transaction Control with Data Transfer Using DAT Line Sequence (Not using DMA)

(1) 실행된 1 블록의 데이터의 바이트 길이에 해당하는 값을 Block Size register 에 설정한다.

(2) 실행된 데이터 블록개수에 해당하는 값을 Block Count register 에 설정한다.

(3) 실행된 command 에 해당하는 값을 Argument register 에 설정한다.

(4) Multi/Signle Block Select 와 Block Count Enalbe 값을 설정한다. 그리고 이 시점에서, 실행한 command 에 해당하는 값을 Direction, Auto CMD12 Enable, DMA Enable 에 설정한다.

(5) 실행된 command 에 해당하는 값을 Command register 에 설정한다.

NOTE: Command register 의 상위 바이트에 writing 할 때, SD command 가 실행된다.

(6) Command Complete Interrupt 를 기다린다.

(7) Normal Interrupt Status register 의 Command Complete(STACMDCMPLT)에 1 을 쓴다, 이 비트를 clear 하기 위해.

(8) Response register 를 읽고 실행된 command 에 따르는 필요한 정보를 가져온다.

(9) 만약 이 절차가 카드에 쓰기를 위한 것이라면, step(10-W)로 계속 수행한다. 카드로부터 읽어오는 경우라면 step(10-R)로 간다.

(10-W) Buffer Write Ready Interrupt 를 기다린다.

(11-W) Normal Interrupt Status register 의 Buffer Write Ready(STABUFWTRDY)에 1 을 쓴다, 이 비트를 clear 하기 위해.

(12-W) Buffer Data Port register 에 block data (step(1)에서 지정한 바이트수와 일치하는)를 쓴다.

(13-W) 모든 블록이 보내질때까지 반복한다. 그런 다음 step(14)로 간다.

(10-R) Buffer Read Ready Interrupt 를 기다린다.

(11-R) Normal Interrupt Status register 의 Buffer Read Ready(STABUFRDRDY)에 1 을 쓴다, 이 비트를 clear 하기 위해.

(12-R) Buffer Data Port register 로부터 block data (step(1)에서 지정한 바이트수와 일치하는)를 읽는다.

(13-R) 모든 블록이 받아질때까지 반복한다. 그런 다음 step(14)로 계속 수행한다.

(14) 만약 이 절차가 Single 또는 Multiple Block Transfer 를 위한 것이라면, step(15)로 계속 진행한다. Infinite Block Transfer 인 경우에는 step(17)로 간다.

(15) Transfer Complete Interrupt 를 기다린다.

(16) Normal Interrupt Status Register 의 Transfer Complete(STATRANCMPLT)에 1 을 쓴다, 이 비트를 clear 하기 위해.

(17) abort transaction(데이터 전송 중단) 절차를 수행한다.

NOTE: step(1)과 step(2)는 동시에 실행될 수 있다. step(4)와 step(5)는 동시에 실행될 수 있다.

27.4.13 USING DMA (DMA 사용)

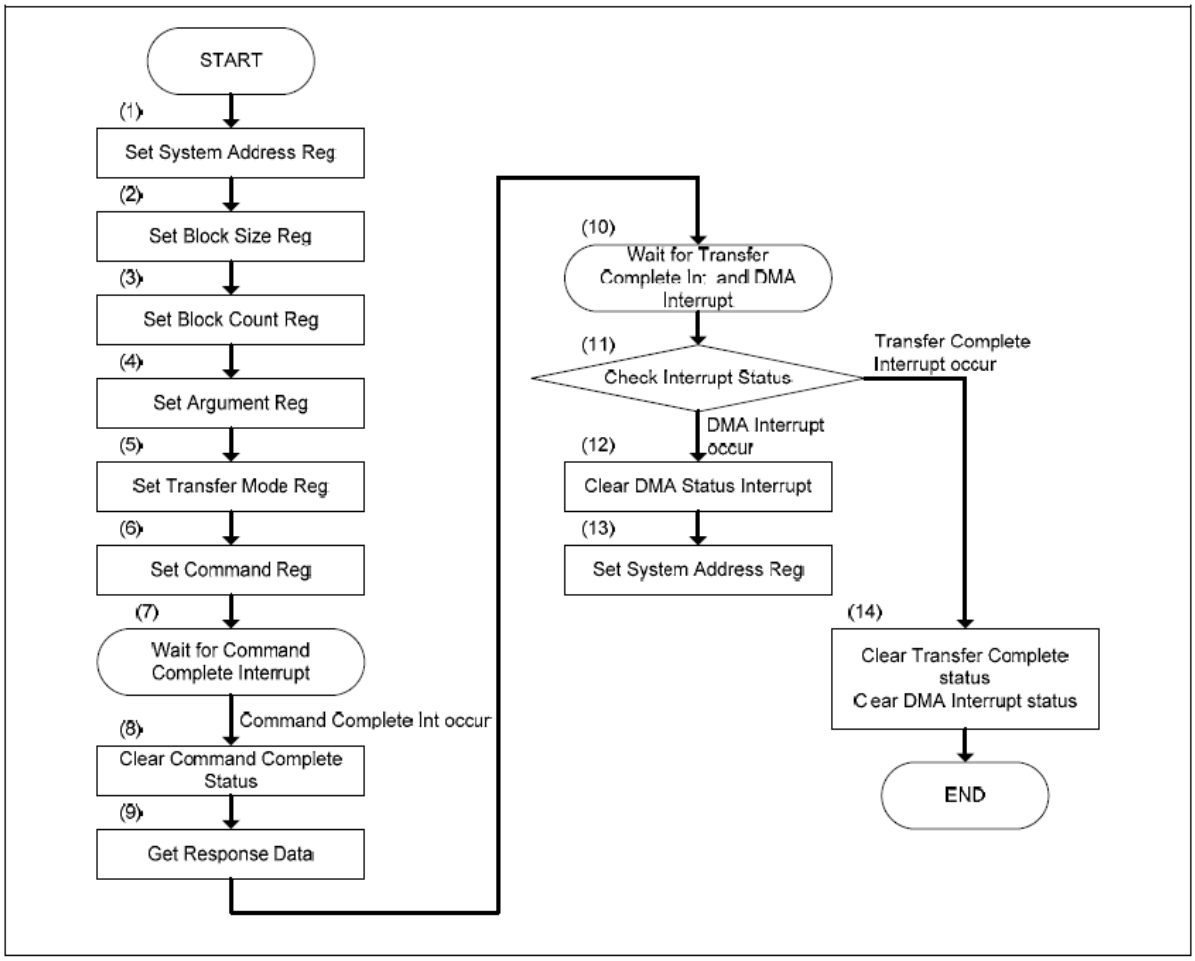

Figure 27-12. Transaction Control with Data Transfer Using DAT Line Sequence (Using DMA)

(1) DMA 사용을 위해 System Address register 의 system address 를 설정한다.

(2) 실행된 블록의 바이트 길이에 해당하는 값을 Block Size register 에 설정한다.

(3) 실행된 블록의 개수에 해당하는 값을 Block Count register(BLKCNT)에 설정한다.

(4) 수행된 command 에 해당하는 값을 Argument register(ARGUMENT)에 설정한다.

(5) Multi/Single Block Select 와 Block Count Enable 에 대한 값을 설정한다.
그리고 이 시점에서 실행된 command 에 해당하는 값을 Data Transfer Direction, Auto CMD12 Enable, DMA Enable 에 설정한다.

(6) 발생된 command 에 대한 값을 Command register(CMDREG)에 설정한다.

NOTE: Command register 의 상위 바이트에 writing 할 때, SD command 가 수행되고 DMA 가 시작된다.

(7) Command Complete Interrupt 를 기다린다.
(8) Normal Interrupt Status register 의 Command Complete(STACMDCMPLT)에 1 을 쓴다, 이 비트를 clear 하기 위해.
(9) Response register 를 읽고 실행된 command 에 해당하는 필요한 정보를 가져온다.
(10) Transfer Complete Interrupt 와 DMA Interrupt 를 기다린다.
(11) 만약 Transfer Complete(STATRANCMPLT)가 1 로 설정되어 있다면, step(14)로 가고, 만약 DMA Interrupt 가 1 로 설정되어 있다면 step(12)로 계속 진행한다. Transfer Complete 는 DMA Interrupt 보다 우선순위가 높다.
(12) Normal Interrupt Status register 의 DMA Interrupt 에 1 을 쓴다, 이 비트를 clear 하기 위해.
(13) 다음 데이터 위치를 위한 next system address 를 System Address register 에 설정하고 step(10)으로 간다.
(14) Normal Interrupt Status register 의 Transfer Complete 와 DMA Interrupt 에 1 을 쓴다, 이 비트를 clear 하기 위해.

NOTE: step(2)와 step(3)은 동시에 실행될 수 있다. step(5)와 step(6) 또한 동시에 실행될 수 있다.

27.5 Abort Transaction (데이터전송 중단)

abort transaction 은 SD memory card 에서는 CMD12(Stop command)를 실행함으로서, SDIO 카드는 CMD52 를 실행함으로서 수행된다. 호스트 드라이버가 abort transaction 을 어디서 필요한지에 따라 2 가지 경우가 있다. 첫 번째 경우는 호스트 드라이버가 Infinite Block Transfer 를 정지했을 때이고, 두 번째 경우는 호스트 드라이버가 Multiple Block Transfer 가 수행되는 동안에 정지할 때이다.

Abort Command 를 수행하는 방법에는 2 가지가 있다. 첫 번째는 비동기 중단(asynchronous abort)이다. 두 번째는 동기적 중단(synchronous abort)이다.

비동기 중단 절차에서호스트 드라이버는 Present State register 의 Command Inhibit(CMD)가 1 로 설정되지 않는한 언제나 수행될 수 있다. 동기적 중단에서는 호스트 드라이버가 Block Gap Control register 의 Stop At Block Gap Request 를 사용해서 데이터 전송이 정지한 다음 Abort Command 를 실행한다.

27.6 DMA Transaction (DMA 데이터 전송)

DMA 는 주변장치가 CPU 의 간섭없이 메모리를 읽고 쓸 수 있게 해준다. 오직 하나의 SD command 트랜잭션만 DMA 에 의해 실행될 수 있다. DMA 를 지원하는 호스트 콘트롤러는 single block 과 multiple block 전송을 모두 지원한다.

System Address register 는 first data address 를 가리키고, 데이터는 그 후 주소로부터 순차적으로 엑세스된다. 호스트 콘트롤러 레지스터는 DMA 전송동안 non-DAT line command 를 수행하기 위해 접근가능상태로 남아있을 것이다. DMA 전송의 결과는 시스템 버스 트랜잭션 사용 방법과 관계없이 동일할 것이다.

DMA 전송은 Block Gap Control register 의 control bit 를 사용함으로서 정지와 재시작을 할 수 있다. Block Gap Request 가 1 로 설정되어서 Stop 일 때, DMA 전송은 중지될 것이다. Continue Request 가 설정되거나 Resume Command 가 실행되었을 때, DMA 는 전송을 수행하는 것을 계속 진행할 것이다. 자세한 사항은 Block Gap Control register 를 참고하자. 만약 SD 버스 에러가 발생했다면, SD 버스 전송은 정지되고 DMA 전송도 중지될 것이다. DAT 라인을 위해 Software Reset register 의 Software Reset 을 설정하는 것은 DMA 전송을 중단시킬 것이다.

27.7 SD/MMC 호스트 콘트롤러 특수 레지스터

27.7.1 CONFIGURATION REGISTER TYPES

Configuration register fields are assigned to one of the attributes described below:

Register Attribute	Description
RO	Read-only register: Register bits are read-only and cannot be altered by software or any reset operation. Writes to these bits are ignored.
ROC	Read-only status: These bits are initialized to zero at reset. Writes to these bits are ignored.
RW or R/W	Read-write register: Register bits are read-write and may be either set or cleared by software to the desired state.
RW1C	Read-only status, Write-1-to-clear status: Register bits indicate status when read, a set bit indicating a status event may be cleared by writing a 1. Writing a 0 to RW1C bits has no effect.
RWAC	Read-Write, automatic clear register: The Host Driver requests a Host Controller operation by setting the bit. The Host Controllers shall clear the bit automatically when the operation is complete. Writing a 0 to RWAC bits has no effect.
HWInit	Hardware Initialized: Register bits are initialized by firmware or hardware mechanisms such as pin strapping or serial EEPROM. Bits are read-only after initialization, and writes to these bits are ignored.
Rsvd or Reserved	Reserved. These bits are initialized to zero, and writes to them are ignored.

27.7.2 SYSTEM ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
SYSAD0	0x7C200000	R/W	System Address register (Channel 0)	0x0
SYSAD1	0x7C300000	R/W	System Address register (Channel 1)	0x0
SYSAD2	0x7C400000	R/W	System Address register (Channel 2)	0x0

This register contains the physical system memory address used for DMA transfers.

Name	Bit	Description	Initial Value
SYSAD	[30:0]	**DMA System Address** This register contains the system memory address for a DMA transfer. When the Host Controller stops a DMA transfer, this register shall point to the system address of the next contiguous data position. It can be accessed only if no transaction is executing (i.e., after a transaction has stopped). Read operations during transfers may return an invalid value. The Host Driver shall initialize this register before starting a DMA transaction. After DMA has stopped, the next system address of the next contiguous data position can be read from this register. The DMA transfer waits at the every boundary specified by the **Host DMA Buffer Boundary** in the *Block Size* register. The Host Controller generates **DMA Interrupt** to request the Host Driver to update this register. The Host Driver set the next system address of the next data position to this register. When the most upper byte of this register (003h) is written, the Host Controller restarts the DMA transfer. When restarting DMA by the Resume command or by setting **Continue Request** in the *Block Gap Control* register, the Host Controller shall start at the next contiguous address stored here in the *System Address* register.	0x00

27.7.3 BLOCK SIZE REGISTER

This register is used to configure the number of bytes in a data block.

Register	Address	R/W	Description	Reset Value
BLKSIZE0	0x7C200004	R/W	Host DMA Buffer Boundary and Transfer Block Size Register (Channel 0)	0x0
BLKSIZE1	0x7C300004	R/W	Host DMA Buffer Boundary and Transfer Block Size Register (Channel 1)	0x0
BLKSIZE2	0x7C400004	R/W	Host DMA Buffer Boundary and Transfer Block Size Register (Channel 2)	0x0

Name	Bit	Description	Initial Value
	[15]	Reserved	0
BUFBOUND	[14:12]	**Host DMA Buffer Boundary** The large contiguous memory space may not be available in the virtual memory system. To perform long DMA transfer, System Address register shall be updated at every system memory boundary during DMA transfer. These bits specify the size of contiguous buffer in the system memory. The DMA transfer shall wait at the every boundary specified by these fields and the Host Controller generates the DMA Interrupt to request the Host Driver to update the System Address register. In case of this register is set to 0 (buffer size = 4K bytes), lower 12-bit of byte address points data in the contiguous buffer and the upper 20-bit points the location of the buffer in the system memory. The DMA transfer stops when the Host Controller detects carry out of the address from bit 11 to 12. These bits shall be supported when the DMA Support in the Capabilities register is set to 1 and this function is active when the DMA Enable in the Transfer Mode register is set to 1. 000b = 4K bytes (Detects A11 carry out) 001b = 8K bytes (Detects A12 carry out) 010b = 16K Bytes (Detects A13 carry out) 011b = 32K Bytes (Detects A14 carry out) 100b = 64K bytes (Detects A15 carry out) 101b = 128K Bytes (Detects A16 carry out) 110b = 256K Bytes (Detects A17 carry out) 111b = 512K Bytes (Detects A18 carry out)	0

Name	Bit	Description	Initial Value
BLKSIZE	[11:0]	**Transfer Block Size** This register specifies the block size of data transfers for CMD17, CMD18, CMD24, CMD25, and CMD53. Values ranging from 1 up to the maximum buffer size can be set. In case of memory, it shall be set up to 512 bytes. It can be accessed only if no transaction is executing (i.e., after a transaction has stopped). Read operations during transfers may return an invalid value, and write operations shall be ignored. 0200h = 512 Bytes 01FFh = 511 Bytes … … 0004h = 4 Bytes 0003h = 3 Bytes 0002h = 2 Bytes 0001h = 1 Byte 0000h = No data transfer	0

27.7.4 BLOCK COUNT REGISTER

This register is used to configure the number of data blocks.

Register	Address	R/W	Description	Reset Value
BLKCNT0	0x7C200006	R/W	Blocks Count For Current Transfer (Channel 0)	0x0
BLKCNT1	0x7C300006	R/W	Blocks Count For Current Transfer (Channel 1)	0x0
BLKCNT2	0x7C400006	R/W	Blocks Count For Current Transfer (Channel 2)	0x0

Name	Bit	Description	Initial Value

BLKCNT	[15:0]	**Blocks Count For Current Transfer** This register is enabled when **Block Count Enable** in the *Transfer Mode* register is set to 1 and is valid only for multiple block transfers. The Host Driver shall set this register to a value between 1 and the maximum block count. The Host Controller decrements the block count after each block transfer and stops when the count reaches zero. Setting the block count to 0 results in no data blocks being transferred. This register must be accessed only when no transaction is executing (i.e., after transactions are stopped). During data transfer, read operations on this register may return an invalid value and write operations are ignored. When saving transfer context as a result of a Suspend command, the number of blocks yet to be transferred can be determined by reading this register. When restoring transfer context prior to issuing a Resume command, the Host Driver shall restore the previously saved block count. FFFFh = 65535 blocks … … 0002h = 2 blocks 0001h = 1 block 0000h = Stop Count	0

27.7.5 ARGUMENT REGISTER

This register contains the SD Command Argument.

Register	Address	R/W	Description	Reset Value
ARGUMENT0	0x7C200008	R/W	Command Argument Register (Channel 0)	0x0
ARGUMENT1	0x7C300008	R/W	Command Argument Register (Channel 1)	0x0
ARGUMENT2	0x7C400008	R/W	Command Argument Register (Channel 2)	0x0

Name	Bit	Description	Initial Value
ARGUMENT	[31:0]	**Command Argument** The SD Command Argument is specified as bit[39:8] of Command-Format in the SD Memory Card Physical Layer Specification.	0

27.7.6 TRANSFER MODE REGISTER

This register is used to control the operation of data transfers. The Host Driver shall set this register before issuing a command which transfers data (Refer to **Data Present Select** in the *Command* register), or before issuing a Resume command. The Host Driver shall save the value of this register when the data transfer is suspended (as a result of a Suspend command) and restore it before issuing a Resume command. To prevent data loss, the Host Controller shall implement write protection for this register during data transactions. Writes to this register shall be ignored when the **Command Inhibit (DAT)** in the *Present State* register is 1.

Register	Address	R/W	Description	Reset Value
TRNMOD0	0x7C20000C	R/W	Transfer Mode Setting Register (Channel 0)	0x0
TRNMOD1	0x7C30000C	R/W	Transfer Mode Setting Register (Channel 1)	0x0
TRNMOD2	0x7C40000C	R/W	Transfer Mode Setting Register (Channel 2)	0x0

Name	Bit	Description	Initial Value
	[15:10]	**Reserved**	0
CCSCON	[9:8]	**Command Completion Signal Control** '00' = No CCS Operation (Normal operation, Not CE-ATA mode) '01' = Read or Write data transfer CCS enable (Only CE-ATA mode) '10' = Without data transfer CCS enable (Only CE-ATA mode) '11' = Abort Completion Signal (ACS) generation (Only CE-ATA mode)	0
	[7:6]	**Reserved**	0
MUL1SIN0	[5]	**Multi / Single Block Select** This bit enables multiple block DAT line data transfers. For any other commands, this bit shall be set to 0. If this bit is 0, it is not necessary to set the *Block Count* register. (Refer to the Table below **"Determination of Transfer Type"**) 1 = Multiple Block 0 = Single Block	0
RD1WT0	[4]	**Data Transfer Direction Select** This bit defines the direction of DAT line data transfers. The bit is set to 1 by the Host Driver to transfer data from the SD card to the SD Host Controller and it is set to 0 for all other commands. 1 = Read (Card to Host) 0 = Write (Host to Card)	0
-	[3]	**Reserved**	0

Name	Bit	Description	Initial Value
ENACMD12	[2]	**Auto CMD12 Enable** Multiple block transfers for memory require CMD12 to stop the transaction. When this bit is set to 1, the Host Controller shall issue CMD12 automatically when last block transfer is completed. The Host Driver shall not set this bit to issue commands that do not require CMD12 to stop data transfer. 1 = Enable 0 = Disable	0
ENBLKCNT	[1]	**Block Count Enable** This bit is used to enable the *Block Count* register, which is only relevant for multiple block transfers. When this bit is 0, the *Block Count* register is disabled, which is useful in executing an infinite transfer. (Refer to the Table below "**Determination of Transfer Type**") 1 = Enable 0 = Disable	0
ENDMA	[0]	**DMA Enable** This bit enables DMA functionality. DMA can be enabled only if it is supported as indicated in the **DMA Support** in the *Capabilities* register. If DMA is not supported, this bit is meaningless and shall always read 0. If this bit is set to 1, a DMA operation shall begin when the Host Driver writes to the upper byte of *Command* register (00Fh). 1 = Enable 0 = Disable	0

Table below shows the summary of how register settings determine types of data transfer.

Determination of Transfer Type

Multi/Single Block Select	Block Count Enable	*Block Count*	Function
0	Don't care	Don't care	Single Transfer
1	0	Don't care	Infinite Transfer
1	1	Not Zero	Multiple Transfer
1	1	Zero	Stop Multiple Transfer

NOTE: For CE-ATA access, (Auto) CMD12 must be issued after Command Completion Signal Disable.

27.7.7 COMMAND REGISTER

This register contains the SD Command Argument.

Register	Address	R/W	Description	Reset Value
CMDREG0	0x7C20000E	R/W	Command Register (Channel 0)	0x0
CMDREG1	0x7C30000E	R/W	Command Register (Channel 1)	0x0
CMDREG2	0x7C40000E	R/W	Command Register (Channel 2)	0x0

The Host Driver shall check the **Command Inhibit (DAT)** bit and **Command Inhibit (CMD)** bit in the *Present State* register before writing to this register. Writing to the upper byte of this register triggers SD command generation. The Host Driver is responsible to write this register because the Host Controller does not protect for writing when **Command Inhibit (CMD)** is set.

Name	Bit	Description	Initial Value
	[15:14]	**Reserved**	
CMDIDX	[13:8]	**Command Index** These bits shall be set to the command number (CMD0-63, ACMD0-63) that is specified in bits 45-40 of the Command-Format in the SD Memory Card Physical Layer Specification and SDIO Card Specification.	
CMDTYP	[7:6]	**Command Type** There are three types of special commands: Suspend, Resume and Abort. These bits **shall** be set to 00b for all other commands. • Suspend Command If the Suspend command succeeds, the Host Controller shall assume the SD Bus has been released and that it is possible to issue the next command, which uses the ***DAT*** line. The Host Controller shall de-assert Read Wait for read transactions and stop checking busy for write transactions. The interrupt cycle shall start, in 4-bit mode. If the Suspend command fails, the Host Controller shall maintain its current state, and the Host Driver shall restart the transfer by setting **Continue Request** in the *Block Gap Control* register. • Resume Command The Host Driver re-starts the data transfer by restoring the registers in the range of 000-00Dh. (Refer to Suspend and Resume mechanism) The Host Controller shall check for busy before starting write transfers. • Abort Command If this command is set when executing a read transfer, the Host Controller shall stop reads to the buffer. If this command is set when executing a write transfer, the Host Controller shall stop driving the ***DAT*** line. After issuing the Abort command, the Host Driver must issue a softwares reset. (Refer to Abort Transaction) 11b = Abort CMD12, CMD52 for writing "I/O Abort" in CCCR 10b = Resume CMD52 for writing "Function Select" in CCCR 01b = Suspend CMD52 for writing "Bus Suspend" in CCCR 00b = Normal Other commands	

Name	Bit	Description	Initial Value
DATAPRNT	[5]	**Data Present Select** This bit is set to 1 to indicate that data is present and shall be transferred using the ***DAT*** line. It is set to 0 for the following: (1) Commands using only ***CMD*** line (ex. CMD52). (2) Commands with no data transfer but using busy signal on ***DAT***[0] line (R1b or R5b ex. CMD38) (3) Resume command 1 = Data Present 0 = No Data Present	
ENCMDIDX	[4]	**Command Index Check Enable** If this bit is set to 1, the Host Controller shall check the Index field in the response to see if it has the same value as the command index. If it is not, it is reported as a Command Index Error. If this bit is set to 0, the Index field is not checked. 1 = Enable 0 = Disable	
ENCMDCRC	[3]	**Command CRC Check Enable** If this bit is set to 1, the Host Controller shall check the CRC field in the response. If an error is detected, it is reported as a Command CRC Error. If this bit is set to 0, the CRC field is not checked. The number of bits checked by the CRC field value changes according to the length of the response. 1 = Enable 0 = Disable	
	[2]	**Reserved**	
RSPTYP	[1:0]	**Response Type Select** 00 = No Response 01 = Response Length 136 10 = Response Length 48 11 = Response Length 48 check Busy after response	

Relation Between Parameters and the Name of Response Type

Response Type	Index Check Enable	CRC Check Enable	Name of Response Type
00	0	0	No Response
01	0	1	R2
10	0	0	R3, R4
10	1	1	R1, R6, R5
11	1	1	R1b, R5b

These bits determine Response types.

NOTES:

1. In the SDIO specification, response type notation of R5b is not defined. R5 includes R5b in the SDIO specification. But R5b is defined in this specification to specify the Host Controller shall check busy after receiving response. For example, usually CMD52 is used as R5 but I/O abort command shall be used as R5b.
2. For CMD52 to read BS after writing "Bus Suspend," Command Type must be "Suspend" as well.

27.7.8 RESPONSE REGISTER

This register is used to store responses from SD cards.

Register	Address	R/W	Description	Reset Value
RSPREG0_0	0x7C200010	ROC	Response Register 0 (Channel 0)	0x0
RSPREG1_0	0x7C200014	ROC	Response Register 1 (Channel 0)	0x0
RSPREG2_0	0x7C200018	ROC	Response Register 2 (Channel 0)	0x0
RSPREG3_0	0x7C20001C	ROC	Response Register 3 (Channel 0)	0x0

Register	Address	R/W	Description	Reset Value
RSPREG0_1	0x7C300010	ROC	Response Register 0 (Channel 1)	0x0
RSPREG1_1	0x7C300014	ROC	Response Register 1 (Channel 1)	0x0
RSPREG2_1	0x7C300018	ROC	Response Register 2 (Channel 1)	0x0
RSPREG3_1	0x7C30001C	ROC	Response Register 3 (Channel 1)	0x0

Register	Address	R/W	Description	Reset Value
RSPREG0_2	0x7C400010	ROC	Response Register 0 (Channel 2)	0x0
RSPREG1_2	0x7C400014	ROC	Response Register 1 (Channel 2)	0x0
RSPREG2_2	0x7C400018	ROC	Response Register 2 (Channel 2)	0x0
RSPREG3_2	0x7C40001C	ROC	Response Register 3 (Channel 2)	0x0

Name	Bit	Description	Initial Value
CMDRSP	[127:0]	**Command Response** The Table below describes the mapping of command responses from the SD Bus to this register for each response type. In the table, R[] refers to a bit range within the response data as transmitted on the SD Bus, REP[] refers to a bit range within the *Response* register. 128-bit Response bit order : {RSPREG3, RSPREG2, RSPREG1, RSPREG0}	

Response Bit Definition for Each Response Type.

Kind of Response	Meaning of Response	Response Field	Response Register
R1, R1b (normal response)	Card Status	R [39:8]	REP [31:0]
R1b (Auto CMD12 response)	Card Status for Auto CMD12	R [39:8]	REP [127:96]
R2 (CID, CSD register)	CID or CSD reg. incl.	R [127:8]	REP [119:0]
R3 (OCR register)	OCR register for memory	R [39:8]	REP [31:0]
R4 (OCR register)	OCR register for I/O etc	R [39:8]	REP [31:0]
R5,R5b	SDIO response	R [39:8]	REP [31:0]
R6 (Published RCA response)	New published RCA[31:16] etc	R [39:8]	REP [31:0]

The Response Field indicates bit positions of "Responses" defined in the PHYSICAL LAYER SPECIFICATION Version 1.01. The Table (upper) shows that most responses with a length of 48 (R[47:0]) have 32 bits of the response data (R[39:8]) stored in the *Response* register at REP[31:0]. Responses of type R1b (Auto CMD12 responses) have response data bits R[39:8] stored in the *Response* register at REP[127:96]. Responses with length 136 (R[135:0]) have 120 bits of the response data (R[127:8]) stored in the *Response* register at REP[119:0].

To be able to read the response status efficiently, the Host Controller only stores part of the response data in the *Response* register. This enables the Host Driver to efficiently read 32 bits of response data in one read cycle on a 32-bit bus system. Parts of the response, the Index field and the CRC, are checked by the Host Controller (as specified by the **Command Index Check Enable** and the **Command CRC Check Enable** bits in the *Command* register) and generate an error interrupt if an error is detected. The bit range for the CRC check depends on the response length. If the response length is 48, the Host Controller shall check R[47:1], and if the response length is 136 the Host Controller shall check R[119:1].

Since the Host Controller may have a multiple block data DAT line transfer executing concurrently with a CMD_wo_DAT command, the Host Controller stores the Auto CMD12 response in the upper bits (REP[127:96]) of the *Response* register. The CMD_wo_DAT response is stored in REP[31:0]. This allows the Host Controller to avoid overwriting the Auto CMD12 response with the CMD_wo_DAT and vice versa.

When the Host Controller modifies part of the *Response* register, as shown in the Table above, it shall preserve the unmodified bits.

27.7.9 BUFFER DATA PORT REGISTER

32-bit data port register to access internal buffer.

Register	Address	R/W	Description	Reset Value
BDATA0	0x7C200020	R/W	Buffer Data Register (Channel 0)	0x0
BDATA1	0x7C300020	R/W	Buffer Data Register (Channel 1)	0x0
BDATA2	0x7C400020	R/W	Buffer Data Register (Channel 2)	0x0

Name	Bit	Description	Initial Value
BUFDAT		**Buffer Data** The Host Controller buffer can be accessed through this 32-bit *Data Port* register.	0

Detailed documents are to be copied from SD Host Standard Specification.

27.7.10 PRESENT STATE REGISTER

This register contains the SD Command Argument.

Register	Address	R/W	Description	Reset Value
PRNSTS0	0x7C200024	RO/ROC	Present State Register (Channel 0)	0x000A0000
PRNSTS1	0x7C300024	RO/ROC	Present State Register (Channel 1)	0x000A0000
PRNSTS2	0x7C400024	RO/ROC	Present State Register (Channel 2)	0x000A0000

Name	Bit	Description	Initial Value
	[31:25]	**Reserved**	0
PRNTCMD	[24]	**CMD Line Signal Level** (RO) This status is used to check the ***CMD*** line level to recover from errors, and for debugging. **Note:** ***CMD*** port is mapped to ***SD0_CMD*** pin	0
PRNTDAT	[23:20]	**DAT[3:0] Line Signal Level** (RO) This status is used to check the ***DAT*** line level to recover from errors, and for debugging. This is especially useful in detecting the busy signal level from ***DAT***[0]. D23 : DAT[3] D22 : DAT[2] D21 : DAT[1] D20 : DAT[0] **Note:** ***DAT*** port is mapped to ***SD0_DAT*** pin	Line State
-	[19]	**Reserved**	1
PRNTCD	[18]	**Card Detect Pin Level** (RO) This bit reflects the inverse value of the ***SDCD#*** pin. Debouncing is not performed on this bit. This bit may be valid when **Card State Stable** is set to 1, but it is not guaranteed because of propagation delay. Use of this bit is limited to testing since it must be debounced by software. 1 = Card present (***SDCD#***=0) 0 = No card present (***SDCD#***=1) **Note:** ***SDCD#*** port is mapped to ***SD0_nCD*** pin, SD2_nCD(Channel 2) port is fixed to LOW.	Line State
STBLCARD	[17]	**Card State Stable** (RO) This bit is used for testing. If it is 0, the **Card Detect Pin Level** is not stable. If this bit is set to 1, it means the **Card Detect Pin Level** is stable. No Card state can be detected by this bit is set to 1 and **Card Inserted** is set to 0. The **Software Reset For All** in the *Software Reset* register shall not affect this bit. 1 = No Card or Inserted 0 = Reset or Debouncing	1 (After Reset)

Name	Bit	Description	Initial Value
INSCARD	[16]	**Card Inserted** (RO) This bit indicates whether a card has been inserted. The Host Controller shall debounce this signal so that the Host Driver will not need to wait for it to stabilize. Changing from 0 to 1 generates a **Card Insertion** interrupt in the *Normal Interrupt Status* register and changing from 1 to 0 generates a **Card Removal** interrupt in the *Normal Interrupt Status* register. The **Software Reset For All** in the *Software Reset* register will not affect this bit. If a card is removed when its power is on and its clock is oscillating, the Host Controller shall clear **SD Bus Power** in the *Power Control* register and **SD Clock Enable** in the *Clock Control* register. When this bit is changed from 1 to 0, the Host Controller shall immediately stop driving ***CMD*** and ***DAT[3:0]*** (tri-state). In addition, the Host Driver must clear the Host Controller by the **Software Reset For All** in *Software Reset* register. The card detect is active regardless of the **SD Bus Power**. 1 = Card Inserted 0 = Reset or Debouncing or No Card	0
	[15:12]	**Reserved**	
BUFRDRDY	[11]	**Buffer Read Enable** (ROC) This status is used for non-DMA read transfers. The Host Controller may implement multiple buffers to transfer data efficiently. This read only flag indicates that valid data exists in the host side buffer status. If this bit is 1, readable data exists in the buffer. A change of this bit from 1 to 0 occurs when all the block data is read from the buffer. A change of this bit from 0 to 1 occurs when block data is ready in the buffer and generates the **Buffer Read Ready** interrupt. 1 = Read enable 0 = Read disable	0

Name	Bit	Description	Initial Value
BUFWTRDY	[10]	**Buffer Write Enable** (ROC) This status is used for non-DMA write transfers. The Host Controller can implement multiple buffers to transfer data efficiently. This read only flag indicates if space is available for write data. If this bit is 1, data can be written to the buffer. A change of this bit from 1 to 0 occurs when all the block data is written to the buffer. A change of this bit from 0 to 1 occurs when top of block data can be written to the buffer and generates the **Buffer Write Ready** interrupt. 1 = Write enable 0 = Write disable	0
RDTRANACT	[9]	**Read Transfer Active** (ROC) This status is used for detecting completion of a read transfer. This bit is set to 1 for either of the following conditions: (1) After the end bit of the read command. (2) When writing a 1 to **Continue Request** in the *Block Gap Control* register to restart a read transfer. This bit is cleared to 0 for either of the following conditions: (1) When the last data block as specified by block length is transferred to the System. (2) When all valid data blocks have been transferred to the System and no current block transfers are being sent as a result of the **Stop At Block Gap Request** being set to 1. **A Transfer Complete** interrupt is generated when this bit changes to 0. 1 = Transferring data 0 = No valid data	0
WTTRANACT	[8]	**Write Transfer Active** (ROC) This status indicates a write transfer is active. If this bit is 0, it means no valid write data exists in the Host Controller. This bit is set in either of the following cases: (1) After the end bit of the write command. (2) When writing a 1 to **Continue Request** in the *Block Gap Control* register to restart a write transfer. This bit is cleared in either of the following cases: (1) After getting the CRC status of the last data block as specified by the transfer count (Single and Multiple) (2) After getting the CRC status of any block where data transmission is about to be stopped by a **Stop At Block Gap Request**. During a write transaction, a **Block Gap Event** interrupt is generated when this bit is changed to 0, as result of the **Stop At Block Gap Request** being set. This status is useful for the Host Driver in determining when to issue commands during write busy. 1 = Transferring data 0 = No valid data	0

Name	Bit	Description	Initial Value
	[7:3]	**Reserved**	0
DATLINEACT	[2]	**DAT Line Active** (ROC) This bit indicates whether one of the ***DAT*** line on SD Bus is in use. **(a) In the case of read transactions** This status indicates if a read transfer is executing on the SD Bus. Change in this value from 1 to 0 between data blocks generates a **Block Gap Event** interrupt in the *Normal Interrupt Status* register. This bit can be set in either of the following cases: (1) After the end bit of the read command. (2) When writing a 1 to **Continue Request** in the *Block Gap Control* register to restart a read transfer. This bit can be cleared in either of the following cases: (1) When the end bit of the last data block is sent from the SD Bus to the Host Controller. (2) When beginning a wait read transfer at a stop at the block gap initiated by a **Stop At Block Gap Request**. The Host Controller will wait at the next block gap by driving Read Wait at the start of the interrupt cycle. If the Read Wait signal is already driven (data buffer cannot receive data), the Host Controller can wait for current block gap by continuing to drive the Read Wait signal. It is necessary to support Read Wait in order to use the suspend / resume function. **(b) In the case of write transactions** This status indicates that a write transfer is executing on the SD Bus. Change in this value from 1 to 0 generates a **Transfer Complete** interrupt in the *Normal Interrupt Status* register. This bit can be set in either of the following cases: (1) After the end bit of the write command. (2) When writing to 1 to **Continue Request** in the *Block Gap Control* register to continue a write transfer. This bit can be cleared in either of the following cases: (1) When the SD card releases write busy of the last data block the Host Controller will detect if output is not busy. If SD card does not drive busy signal for 8 SD Clocks, the Host Controller will consider the card drive "Not Busy". (2) When the SD card releases write busy prior to waiting for write transfer as a result of a **Stop At Block Gap Request**. 1 = DAT Line Active 0 = DAT Line Inactive	0

Name	Bit	Description	Initial Value
CMDIN HDAT	[1]	**Command Inhibit (DAT)** (ROC) (ROC) This status bit is generated if either the **DAT Line Active** or the **Read Transfer Active** is set to 1. If this bit is 0, it indicates the Host Controller can issue the next SD Command. Commands with busy signal belong to **Command Inhibit (DAT)** (ex. R1b, R5b type). Changing from 1 to 0 generates a **Transfer Complete** interrupt in the *Normal Interrupt Status* register. **Note:** The SD Host Driver can save registers in the range of 000-00Dh for a suspend transaction after this bit has changed from 1 to 0. 1 = Cannot issue command which uses the ***DAT*** line 0 = Can issue command which uses the ***DAT*** line	0
CMDIN HCMD	[0]	**Command Inhibit (CMD)** (ROC) If this bit is 0, it indicates the ***CMD*** line is not in use and the Host Controller can issue a SD Command using the ***CMD*** line. This bit is set immediately after the *Command* register (00Fh) is written. This bit is cleared when the command response is received. Even if the **Command Inhibit (DAT)** is set to 1, Commands using only the ***CMD*** line can be issued if this bit is 0. Changing from 1 to 0 generates a **Command Complete** interrupt in the *Normal Interrupt Status* register. If the Host Controller cannot issue the command because of a command conflict error (Refer to **Command CRC Error**) or because of **Command Not Issued By Auto CMD12 Error**, this bit shall remain 1 and the **Command Complete** is not set. Status issuing Auto CMD12 is not read from this bit. 1 = Cannot issue command 0 = Can issue command using only ***CMD*** line	0

NOTE: Buffer Write Enable in Present register must not be asserted for DMA transfers since it generates Buffer Write Ready interrupt

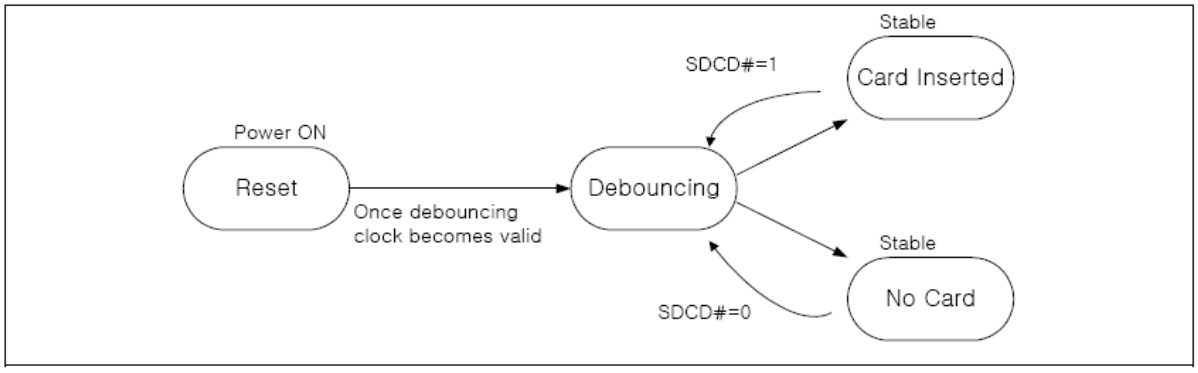

Figure 27-13. Card Detect State

The above Figure 27-13 shows the state definitions of hardware that handles "Debouncing".

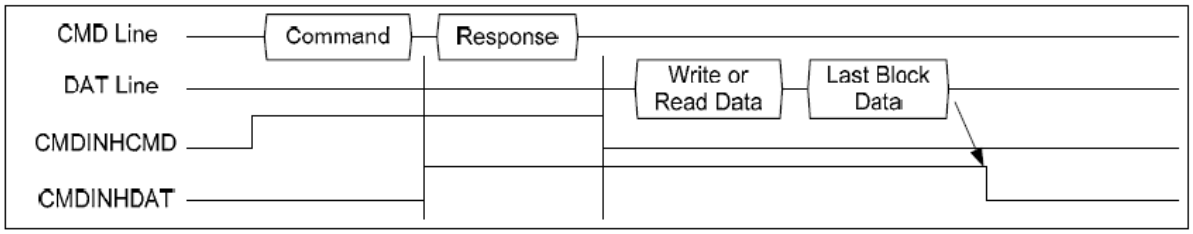

Figure 27-14. Timing of Command Inhibit (DAT) and Command Inhibit (CMD) with data transfer

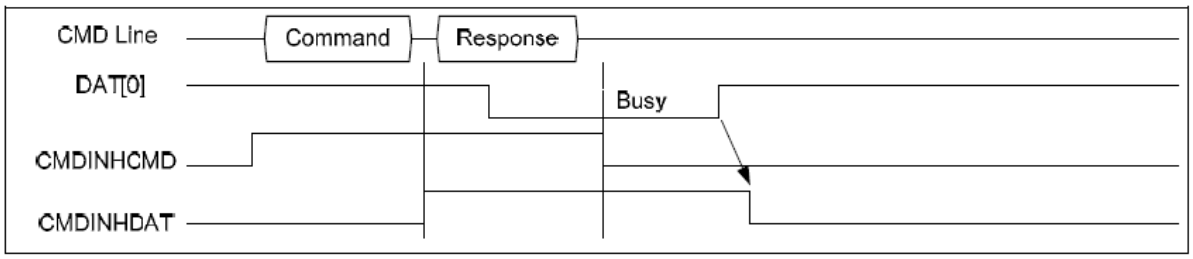

Figure 27-15. Timing of Command Inhibit (DAT) for the case of response with busy

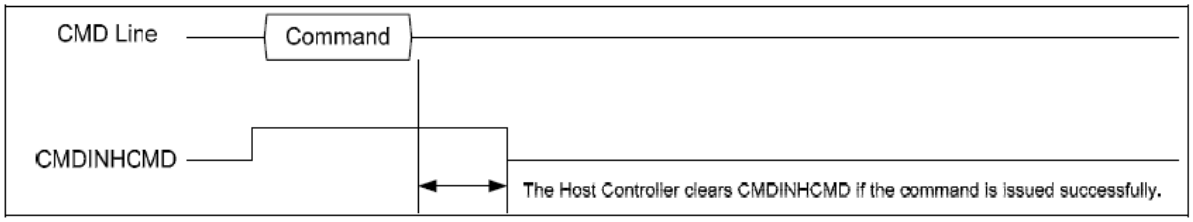

Figure 27-16. Timing of Command Inhibit (CMD) for the case of no response command

27.7.11 HOST CONTROL REGISTER

This register contains the SD Command Argument.

Register	Address	R/W	Description	Reset Value
HOSTCTL0	0x7C200028	R/W	Present State Register (Channel 0)	0x0
HOSTCTL1	0x7C300028	R/W	Present State Register (Channel 1)	0x0
HOSTCTL2	0x7C400028	R/W	Present State Register (Channel 2)	0x0

Name	Bit	Description	Initial Value
CDSIG SEL	[7]	**Card Detect Signal Selection** This bit selects source for the card detection. '1' = The Card Detect Test Level is selected (for test purpose) '0' = SDCD# is selected (for normal use)	0
CDTE STLVL	[6]	**Card Detect Test Level** This bit is enabled while the Card Detect Signal Selection is set to 1 and it indicates card inserted or not. '1' = Card Inserted '0' = No Card	0
WIDE8	[5]	**Extended Data Transfer Width** (It is for MMC 8-bit card.) '1' = 8-bit operation '0' = the bit width is designated by the bit 1 (Data Transfer Width)	0
	[4:3]	**Reserved**	0
ENHIG HSPD	[2]	**High Speed Enable** This bit is optional. Before setting this bit, the Host Driver shall check the **High Speed Support** in the *Capabilities* register. If this bit is set to 0 (default), the Host Controller outputs ***CMD*** line and ***DAT*** lines at the falling edge of the SD Clock (up to 25MHz). If this bit is set to 1, the Host Controller outputs ***CMD*** line and ***DAT*** lines at the rising edge of the SD Clock (up to 50MHz). '1' = High Speed mode '0' = Normal Speed mode	0
WIDE4	[1]	**Data Transfer Width** This bit selects the data width of the Host Controller. The Host Driver shall set it to match the data width of the SD card. '1' = 4-bit mode '0' = 1-bit mode	0
-	[0]	**Reserved**	0

NOTE: Card Detect Pin Level does not simply reflect SDCD# pin, but selects from SDCD, DAT[3], or CDTestlvl depending on CDSSigSel and SDCDSel values.

27.7.12 POWER CONTROL REGISTER

This register contains the SD Command Argument.

Register	Address	R/W	Description	Reset Value
PWRCON0	0x7C200029	R/W	Present State Register (Channel 0)	0x0
PWRCON1	0x7C300029	R/W	Present State Register (Channel 1)	0x0
PWRCON2	0x7C400029	R/W	Present State Register (Channel 2)	0x0

Name	Bit	Description	Initial Value
	[7:4]	Reserved	
SELPWR LVL	[3:1]	**SD Bus Voltage Select** By setting these bits, the Host Driver selects the voltage level for the SD card. Before setting this register, the Host Driver will check the **Voltage Support** bits in the *Capabilities* register. If an unsupported voltage is selected, the Host System will not supply SD Bus voltage. '111b' = 3.3V (Typ.) '110b' = 3.0V (Typ.) '101b' = 1.8V (Typ.) '100b' – '000b' = Reserved	0
PWRON	[0]	**SD Bus Power** Before setting this bit, the SD Host Driver will set **SD Bus Voltage Select**. If the Host Controller detects the No Card state, this bit will be cleared. If this bit is cleared, the Host Controller will immediately stop driving ***CMD*** and ***DAT[3:0]*** (tri-state) and drive ***SDCLK*** to low level. '1' = Power on '0' = Power off	0

27.7.13 BLOCK GAP CONTROL REGISTER

This register contains the SD Command Argument.

Register	Address	R/W	Description	Reset Value
BLKGAP0	0x7C20002A	R/W	Block Gap Control Register (Channel 0)	0x0
BLKGAP1	0x7C30002A	R/W	Block Gap Control Register (Channel 1)	0x0
BLKGAP2	0x7C40002A	R/W	Block Gap Control Register (Channel 2)	0x0

Name	Bit	Description	Initial Value
	[7:4]	Reserved	0
ENINTBGAP	[3]	**Interrupt At Block Gap** This bit is valid only in 4-bit mode of the SDIO card and selects a sample point in the interrupt cycle. Setting to 1 enables interrupt detection at the block gap for a multiple block transfer. Setting to 0 disables interrupt detection during a multiple block transfer. If the SD card cannot signal an interrupt during a multiple block transfer, this bit must be set to 0. When the Host Driver detects an SD card insertion, it will set this bit according to the CCCR of the SDIO card. (RW) '1' = Enabled, '0' = Disabled Note) Interrupt at Block Gap operation is not supported in S3C6400X controller, it should be fixed to 0.	0
ENRWAIT	[2]	**Read Wait Control** The read wait function is optional for SDIO cards. If the card supports read wait, set this bit to enable use of the read wait protocol to stop read data using the ***DAT***[2] line. Otherwise the Host Controller has to stop the SD Clock to hold read data, which restricts commands generation. When the Host Driver detects an SD card insertion, it will set this bit according to the CCCR of the SDIO card. If the card does not support read wait, this bit will never be set to 1 otherwise ***DAT*** line conflict may occur. If this bit is set to 0, Suspend/Resume cannot be supported. (RW) '1' = Enable Read Wait Control, '0' = Disable Read Wait Control	0
CONTREQ	[1]	**Continue Request** This bit is used to restart a transaction, which was stopped using the **Stop At Block Gap Request**. To cancel stop at the block gap, set **Stop At Block Gap Request** to 0 and set this bit 1 to restart the transfer. The Host Controller automatically clears this bit in either of the following cases: (1) If a read transaction, the **DAT Line Active** changes from 0 to 1 as a read transaction restarts. (2) If a write transaction, the **Write Transfer Active** changes from 0 to 1 as the write transaction restarts. Therefore it is not necessary for Host Driver to set this bit to 0. If **Stop At Block Gap Request** is set to 1, any write to this bit is ignored. (RWAC) '1' = Restart, '0' = Not affect	0

Name	Bit	Description	Initial Value
STOP BGAP	[0]	**Stop At Block Gap Request** This bit is used to stop executing a transaction at the next block gap for both DMA and non-DMA transfers. Until the **Transfer Complete** is set to 1, indicating a transfer completion the Host Driver will leave this bit set to 1. Clearing both the **Stop At Block Gap Request** and **Continue Request** will not cause the transaction to restart. Read Wait is used to stop the read transaction at the block gap. The Host Controller shall honour **Stop At Block Gap Request** for write transfers, but for read transfers it requires that the SD card support Read Wait. Therefore the Host Driver does not set this bit during read transfers unless the SD card supports Read Wait and has set **Read Wait Control** to 1. In the case of write transfers in which the Host Driver writes data to the *Buffer Data Port* register, the Host Driver sets this bit after all block data is written. If this bit is set to 1, the Host Driver does not write data to *Buffer Data Port* register. This bit affects **Read Transfer Active**, **Write Transfer Active**, **DAT Line Active** and **Command Inhibit (DAT)** in the *Present State* register. Regarding detailed control of bits D01 and D00. (RW) '1' = Stop '0' = Transfer	0

There are three cases to restart the transfer after stop at the block gap. Which case is appropriate depends on whether the Host Controller issues a Suspend command or the SD card accepts the Suspend command.

Cases are as follows:

(1) If the Host Driver does not issue a Suspend command, the **Continue Request** can be used to restart the transfer.

(2) If the Host Driver issues a Suspend command and the SD card accepts it, a Resume command is used to restart the transfer.

(3) If the Host Driver issues a Suspend command and the SD card does not accept it, the **Continue Request** is used to restart the transfer.

Any time **Stop At B**lock Gap Request stops the data transfer, the Host Driver will wait for Transfer Complete (in the *Normal Interrupt Status* register) before attempting to restart the transfer. When the data transfer by Continue Request is restarted, the Host Driver clears **Stop At Block Gap Request** before or simultaneously.

NOTE:

After setting **Stop At Block Gap Request** field, it must not be cleared unless Block Gap Event or Transfer Complete interrupt occurs. Otherwise, the module hangs.

27.7.14 WAKEUP CONTROL REGISTER

This register is mandatory for the Host Controller, but wakeup functionality depends on the Host Controller system hardware and software. The Host Driver maintains voltage on the SD Bus, by setting **SD Bus Power** to 1 in the *Power Control* register, when wakeup event via Card Interrupt is desired.

Register	Address	R/W	Description	Reset Value
WAKCON0	0x7C20002B	R/W	Wakeup Control Register (Channel 0)	0x0
WAKCON1	0x7C30002B	R/W	Wakeup Control Register (Channel 1)	0x0
WAKCON2	0x7C40002B	R/W	Wakeup Control Register (Channel 2)	0x0

Name	Bit	Description	Initial Value
	[7:3]	Reserved	0
ENWKUPREM	[2]	**Wakeup Event Enable On SD Card Removal** This bit enables wakeup event via **Card Removal** assertion in the *Normal Interrupt Status* register. **FN_WUS** (Wake Up Support) in CIS does not affect this bit. (RW) '1' = Enable '0' = Disable	0
ENWKUPINS	[1]	**Wakeup Event Enable On SD Card Insertion** This bit enables wakeup event via **Card Insertion** assertion in the *Normal Interrupt Status* register. **FN_WUS** (Wake Up Support) in CIS does not affect this bit. (RW) '1' = Enable '0' = Disable	0
ENWKUPINT	[0]	**Wakeup Event Enable On Card Interrupt** This bit enables wakeup event via **Card Interrupt** assertion in the *Normal Interrupt Status* register. This bit can be set to 1 if **FN_WUS** (Wake Up Support) in CIS is set to 1. (RW) '1' = Enable '0' = Disable	0

27.7.15 CLOCK CONTROL REGISTER

At the initialization of the Host Controller, the Host Driver sets the **SDCLK Frequency Select** according to the *Capabilities* register.

Register	Address	R/W	Description	Reset Value
CLKCON0	0x7C20002C	R/W	Command Register (Channel 0)	0x0
CLKCON1	0x7C30002C	R/W	Command Register (Channel 1)	0x0
CLKCON2	0x7C40002C	R/W	Command Register (Channel 2)	0x0

Name	Bit	Description	Initial Value
SELFREQ	[15:8]	**SDCLK Frequency Select** This register is used to select the frequency of ***SDCLK*** pin. The frequency is not programmed directly; rather this register holds the divisor of the **Base Clock Frequency For SD Clock** in the *Capabilities* register. Only the following settings are allowed. 80h: base clock divided by 256 40h: base clock divided by 128 20h: base clock divided by 64 10h: base clock divided by 32 08h: base clock divided by 16 04h: base clock divided by 8 02h: base clock divided by 4 01h: base clock divided by 2 00h: base clock (10MHz-63MHz) Setting 00h specifies the highest frequency of the SD Clock. Setting multiple bits, the most significant bit is used as the divisor. But multiple bits must not be set. The two default divider values can be calculated by the frequency that is defined by the **Base Clock Frequency For SD Clock** in the *Capabilities* register. (1) 25MHz divider value (2) 400kHz divider value According to the SD Physical Specification Version 1.01 and the SDIO Card Specification Version 1.0, maximum SD Clock frequency is 25MHz, and never exceeds this limit. The frequency of SDCLK is set by the following formula: Clock Frequency = (Base Clock) / divisor Therefore, select the smallest possible divisor which results in a clock frequency that is less than or equal to the target frequency. For example, if the **Base Clock Frequency For SD Clock** in the *Capabilities* register has the value 33MHz, and the target frequency is 25MHz, then selecting the divisor value of 01h will yield 16.5MHz, which is the nearest frequency less than or equal to the target. Similarly, to approach a clock value of 400kHz, the divisor value of 40h yields the optimal clock value of 258kHz.	0

Name	Bit	Description	Initial Value
	[7:4]	**Reserved**	
-	[3]	**Reserved**	0
ENSDCLK	[2]	**SD Clock Enable** The Host Controller stops ***SDCLK*** when writing this bit to 0. **SDCLK Frequency Select** can be changed when this bit is 0. Then, the Host Controller shall maintain the same clock frequency until ***SDCLK*** is stopped (Stop at ***SDCLK***=0). If the **Card Inserted** in the *Present State register* is cleared, this bit will be cleared. **(RW)** '1' = Enable '0' = Disable	0
STBLINTCLK	[1]	**Internal Clock Stable** This bit is set to 1 when SD Clock is stable after writing to **Internal Clock Enable** in this register to 1. The SD Host Driver shall wait to set **SD Clock Enable** until this bit is set to 1. **Note:** This is useful when using PLL for a clock oscillator that requires setup time. **(ROC)** '1' = Ready '0' = Not Ready	0
ENINTCLK	[0]	**Internal Clock Enable** This bit is set to 0 when the Host Driver is not using the Host Controller or the Host Controller awaits a wakeup interrupt. The Host Controller must stop its internal clock to go very low power state. Still, registers shall be able to be read and written. Clock starts to oscillate when this bit is set to 1. When clock oscillation is stable, the Host Controller can be set **Internal Clock Stable** in this register to 1. This bit shall not affect card detection. **(RW)** '1' = Oscillate '0' = Stop	

27.7.16 TIMEOUT CONTROL REGISTER

At the initialization of the Host Controller, the Host Driver can set the **Data Timeout Counter Value** according to the *Capabilities* register.

Register	Address	R/W	Description	Reset Value
TIMEOUTCON0	0x7C20002E	R/W	Timeout Control Register (Channel 0)	0x0
TIMEOUTCON1	0x7C30002E	R/W	Timeout Control Register (Channel 1)	0x0
TIMEOUTCON2	0x7C40002E	R/W	Timeout Control Register (Channel 2)	0x0

Name	Bit	Description	Initial Value
	[7:4]	Reserved	0
TIMEOUTCON	[3:0]	**Data Timeout Counter Value** This value determines the interval by which DAT line timeouts are detected. Refer to the **Data Timeout Error** in the *Error Interrupt Status* register for information on factors that dictate timeout generation. Timeout clock frequency will be generated by dividing the base clock TMCLK value by this value. When setting this register, prevent inadvertent timeout events by clearing the **Data Timeout Error Status Enable** (in the *Error Interrupt tatus Enable* register) 1111b Reserved 1110b TMCLK x 2^{27} 1101b TMCLK x 2^{26} …… … 0001b TMCLK x 2^{14} 0000b TMCLK x 2^{13}	0

27.7.17 SOFTWARE RESET REGISTER

A reset pulse is generated when writing 1 to each bit of this register. After completing the reset, the Host Controller clears each bit. Because it takes some time to complete software reset, the SD Host Driver shall confirm that these bits are 0.

Register	Address	R/W	Description	Reset Value
SWRST0	0x7C20002F	R/W	Software Reset Register (Channel 0)	0x0
SWRST1	0x7C30002F	R/W	Software Reset Register (Channel 1)	0x0
SWRST2	0x7C40002F	R/W	Software Reset Register (Channel 2)	0x0

Name	Bit	Description	Initial Value
	[7:3]	Reserved	0
RSTDAT	[2]	**Software Reset For DAT Line** Only part of data circuit is reset. DMA circuit is also reset. (RWAC) The following registers and bits are cleared by this bit: *Buffer Data Port* register Buffer is cleared and initialized. *Present State* register **Buffer Read Enable** **Buffer Write Enable** **Read Transfer Active** **Write Transfer Active** **DAT Line Active** **Command Inhibit (DAT)** *Block Gap Control* register **Continue Request** **Stop At Block Gap Request** *Normal Interrupt Status* register **Buffer Read Ready** **Buffer Write Ready** **DMA Interrupt** **Block Gap Event** **Transfer Complete** '1' = Reset '0' = Work	0
RSTCMD	[1]	**Software Reset For CMD Line** Only part of command circuit is reset. (RWAC). The following registers and bits are cleared by this bit: *Present State* register **Command Inhibit (CMD)** *Normal Interrupt Status* register **Command Complete** '1' = Reset '0' = Work	0

Name	Bit	Description	Initial Value
RSTALL	[0]	**Software Reset For All** This reset affects the entire Host Controller except for the card detection circuit. Register bits of type ROC, RW, RW1C, RWAC, HWInit are cleared to 0. During its initialization, the Host Driver sets this bit to 1 to reset the Host Controller. The Host Controller reset this bit to 0 when capabilities registers are valid and the Host Driver can read them. If this bit is set to 1, the SD card shall reset itself and must be reinitialized by the Host Driver. (RWAC) '1' = Reset '0' = Work	0

27.7.18 NORMAL INTERRUPT STATUS REGISTER

The *Normal Interrupt Status Enable* affects reads of this register, but *Normal Interrupt Signal Enable* does not affect these reads. An interrupt is generated when the Normal Interrupt Signal Enable is enabled and at least one of the status bits is set to 1. For all bits except **Card Interrupt** and **Error Interrupt**, writing 1 to a bit clears it; writing to 0 keeps the bit unchanged. More than one status can be cleared with a single register write. The **Card Interrupt** is cleared when the card stops asserting the interrupt; that is, when the Card Driver services the interrupt condition.

Register	Address	R/W	Description	Reset Value
NORINTSTS0	0x7C200030	ROC/RW1C	Normal Interrupt Status Register (Channel 0)	0x0
NORINTSTS1	0x7C300030	ROC/RW1C	Normal Interrupt Status Register (Channel 1)	0x0
NORINTSTS2	0x7C400030	ROC/RW1C	Normal Interrupt Status Register (Channel 2)	0x0

Name	Bit	Description	Initial Value
STAERR	[15]	**Error Interrupt** If any of the bits in the Error Interrupt Status register are set, then this bit is set. Therefore the Host Driver can efficiently test for an error by checking this bit first. This bit is read only. (ROC) '0' = No Error '1' = Error	0
STAFIA3	[14]	**FIFO SD Address Pointer Interrupt 3 Status (RW1C)** '0' = Occurred '1' = Not Occurred When the FIFO Address of the SD clock side reaches the FIFO Interrupt Address register 3 values, this status bit is asserted.	0
STAFIA2	[13]	**FIFO SD Address Pointer Interrupt 2 Status (RW1C)** '0' = Occurred '1' = Not Occurred When the FIFO Address of the SD clock side reaches the FIFO Interrupt Address register 2 values, this status bit is asserted.	0

Name	Bit	Description	Initial Value
STAFIA1	[12]	**FIFO SD Address Pointer Interrupt 1 Status (RW1C)** '0' = Occurred '1' = Not Occurred When the FIFO Address of the SD clock side reaches the FIFO Interrupt Address register 1 value, this status bit is asserted.	0
STAFIA0	[11]	**FIFO SD Address Pointer Interrupt 0 Status (RW1C)** '0' = Occurred '1' = Not Occurred When the FIFO Address of the SD clock side reaches the FIFO Interrupt Address register 0 value, this status bit is asserted.	0
STARWAIT	[10]	**Read Wait Interrupt Status (RW1C)** '0' = Read Wait Interrupt Occurred '1' = Read Wait Interrupt Not Occurred Note1) After checking response for the suspend command, release Read Wait interrupt status manually if BS = 0 (BS means 'Bus Status' field 'Bus Suspend' register in the SDIO card spec) Note2) Read Wait operation procedure is started after 4-SDCLK from the end of the block data read transfer.	0
STACCS	[9]	**CCS Interrupt Status (RW1C)** Command Complete Signal Interrupt Status bit is for CE-ATA interface mode. '0' = CCS Interrupt Occurred, '1' = CCS Interrupt Not Occurred	0
STACARDINT	[8]	Card Interrupt Writing this bit to 1 does not clear this bit. It is cleared by resetting the SD card interrupt factor. In 1-bit mode, the Host Controller shall detect the Card Interrupt without SD Clock to support wakeup. In 4-bit mode, the card interrupt signal is sampled during the interrupt cycle, so there are some sample delays between the interrupt signal from the SD card and the interrupt to the Host System. It is necessary to define how to handle this delay. When this status has been set and the Host Driver needs to start this interrupt service, Card Interrupt Signal Enable in the Normal Interrupt Signal Enable register must be set to 0 in order to clear the card interrupt status latched in the Host Controller and to stop driving the interrupt signal to the Host System. After completion of the card interrupt service (It must reset interrupt factors in the SD card and the interrupt signal may not be asserted), write to one clear to this register field(RW1C) and set Card Interrupt Signal Enable to 1 to re-start sampling the interrupt signal. The Card Interrupt Status Enable must be remain set to high. (RW1C) Note2,3 '1' = Generate Card Interrupt '0' = No Card Interrupt	0

Name	Bit	Description	Initial Value
STACARDREM	[7]	**Card Removal** This status is set if the Card Inserted in the Present State register changes from 1 to 0. When the Host Driver writes this bit to 1 to clear this status, the status of the Card Inserted in the Present State register must be confirmed. Because the card detect state may possibly be changed when the Host Driver clear this bit and interrupt event may not be generated. (RW1C) '1' = Card removed '0' = Card state stable or Debouncing	0
STACARDINS	[6]	**Card Insertion** This status is set if the Card Inserted in the Present State register changes from 0 to 1. When the Host Driver writes this bit to 1 to clear this status, the status of the Card Inserted in the Present State register must be confirmed. Because the card detect state may possibly be changed when the Host Driver clear this bit and interrupt event may not be generated. (RW1C) '1' = Card inserted '0' = Card state stable or Debouncing	0
STABUFRDRDY	[5]	**Buffer Read Ready** This status is set if the Buffer Read Enable changes from 0 to 1. Refer to the Buffer Read Enable in the Present State register. (RW1C) '1' = Ready to read buffer '0' = Not ready to read buffer	0
STABUFWTRDY	[4]	**Buffer Write Ready** This status is set if the Buffer Write Enable changes from 0 to 1. Refer to the Buffer Write Enable in the Present State register. (RW1C) '1' = Ready to write buffer '0' = Not ready to write buffer	0
STADMAINT	[3]	**DMA Interrupt** This status is set if the Host Controller detects the Host DMA Buffer boundary during transfer. Refer to the Host DMA Buffer Boundary in the Block Size register. Other DMA interrupt factors may be added in the future. This interrupt cannot be generated after the Transfer Complete. (RW1C) '1' = DMA Interrupt is generated '0' = No DMA Interrupt	0
STABLKGAP	[2]	**Block Gap Event** If the Stop At Block Gap Request in the Block Gap Control register is set, this bit is set when both a read / write transaction is stopped at a block gap. If Stop At Block Gap Request is not set to 1, this bit is not set to 1. (RW1C) (1) In the case of a Read Transaction This bit is set at the falling edge of the DAT Line Active Status (When the transaction is stopped at SD Bus timing. The Read Wait must be supported in order to use this function. (2) Case of Write Transaction This bit is set at the falling edge of Write Transfer Active Status (After getting CRC status at SD Bus timing). '1' = Transaction stopped at block gap '0' = No Block Gap Event	0

<table>
<tr><th>Name</th><th>Bit</th><th>Description</th><th>Initial Value</th></tr>
<tr><td>STATRANCMPLT</td><td>[1]</td><td>Transfer Complete
This bit is set when a read / write transfer is completed.
(1) In the case of a Read Transaction
This bit is set at the falling edge of Read Transfer Active Status. There are two cases in which this interrupt is generated. The first is when a data transfer is completed as specified by data length (After the last data has been read to the Host System). The second is when data has stopped at the block gap and completed the data transfer by setting the Stop At Block Gap Request in the Block Gap Control register (After valid data has been read to the Host System).
(2) In the case of a Write Transaction
This bit is set at the falling edge of the DAT Line Active Status. There are two cases in which this interrupt is generated. The first is when the last data is written to the SD card as specified by data length and the busy signal released. The second is when data transfers are stopped at the block gap by setting Stop At Block Gap Request in the Block Gap Control register and data transfers completed. (After valid data is written to the SD card and the busy signal released). (RW1C)
The table below shows that Transfer Complete has higher priority than Data Timeout Error. If both bits are set to 1, the data transfer can be considered complete. Relation between Transfer Complete and Data
<table>
<tr><th>Transfer Complete</th><th>Data Timeout Error</th><th>Meaning of the status</th></tr>
<tr><td>0</td><td>0</td><td>Interrupted by another factor</td></tr>
<tr><td>0</td><td>1</td><td>Timeout occur during transfer</td></tr>
<tr><td>1</td><td>Don’t care</td><td>Data transfer complete</td></tr>
</table>
'1' = Data Transfer Complete
'0' = No transfer complete</td><td>0</td></tr>
<tr><td>STACMDCMPLT</td><td>[0]</td><td>Command Complete
This bit is set when get the end bit of the command response. (Except Auto CMD12) Refer to Command Inhibit (CMD) in the Present State register.
The table below shows that Command Timeout Error has higher priority than Command Complete. If both bits are set to 1, it can be considered that the response was not received correctly. (RW1C)
<table>
<tr><th>Command Complete</th><th>Command Timeout Error</th><th>Meaning of the status</th></tr>
<tr><td>0</td><td>0</td><td>Interrupted by another factor</td></tr>
<tr><td>Don’t care</td><td>1</td><td>Response not received within 64 SDCLK cycles.</td></tr>
<tr><td>1</td><td>0</td><td>Response received</td></tr>
</table>
‘1’ = Command Complete
‘0’ = No command complete</td><td>0</td></tr>
</table>

NOTES:

1. Host Driver may check if interrupt is actually cleared by polling or monitoring the INTREQ port. If HCLK is much faster than SDCLK, it takes long time to be cleared for the bits actually.
2. Card Interrupt status bit keeps previous value until next card interrupt period (level interrupt) and can be cleared when write to 1 (RW1C).
3. SD/MMC Controller of the S3C6400 does not support "card interrupt at block gap" used when the multiple block 4-bit operation.

27.7.19 ERROR INTERRUPT STATUS REGISTER

Signals defined in this register can be enabled by the *Error Interrupt Status Enable* register, but not by the *Error Interrupt Signal Enable* register. The interrupt is generated when the *Error Interrupt Signal Enable* is enabled and at least one of the statuses is set to 1. Writing to 1 clears the bit and writing to 0 keeps the bit unchanged. More than one status can be cleared at the one register write.

Register	Address	R/W	Description	Reset Value
ERRINTSTS0	0x7C200032	ROC/RW1C	Error Interrupt Status Register (Channel 0)	0x0
ERRINTSTS1	0x7C300032	ROC/RW1C	Error Interrupt Status Register (Channel 1)	0x0
ERRINTSTS2	0x7C400032	ROC/RW1C	Error Interrupt Status Register (Channel 2)	0x0

Name	Bit	Description	Initial Value
	[15:9]	**Reserved**	0
STAACMDERR	[8]	**Auto CMD12 Error** Occurs when detecting that one of the bits in *Auto CMD12 Error Status* register has changed from 0 to 1. This bit is set to 1, not only when the errors in Auto CMD12 occur but also when Auto CMD12 is not executed due to the previous command error. '1' = Error '0' = No Error	0
STACURERR	[7]	**Current Limit Error** Not implemented in this version. Always 0.	0
STADENDERR	[6]	**Data End Bit Error** Occurs either when detecting 0 at the end bit position of read data which uses the ***DAT*** line or at the end bit position of the CRC Status. '1' = Error '0' = No Error	0
STADATCRCERR	[5]	**Data CRC Error** Occurs when detecting CRC error when transferring read data which uses the ***DAT*** line or when detecting the Write CRC status having a value of other than "010". '1' = Error '0' = No Error	0
STADATTOUTERR	[4]	**Data Timeout Error** Occurs when detecting one of following timeout conditions. (1) Busy timeout for R1b, R5b type (2) Busy timeout after Write CRC status (3) Write CRC Status timeout (4) Read Data timeout. '1' = Timeout '0' = No Error	0

Name	Bit	Description	Initial Value
STACMDIDXERR	[3]	**Command Index Error** Occurs if a Command Index error occurs in the command response. '1' = Error '0' = No Error	0
STACMDEBITERR	[2]	**Command End Bit Error** Occurs when detecting that the end bit of a command response is 0. '1' = End bit Error generated '0' = No Error	
STACMDCRCERR	[1]	**Command CRC Error** Command CRC Error is generated in two cases. (1) If a response is returned and the **Command Timeout Error** is set to 0 (indicating no timeout), this bit is set to 1 when detecting a CRC error in the command response. (2) The Host Controller detects a ***CMD*** line conflict by monitoring the ***CMD*** line when a command is issued. If the Host Controller drives the ***CMD*** line to 1 level, but detects 0 levels on the ***CMD*** line at the next SDCLK edge, then the Host Controller will abort the command (Stop driving ***CMD*** line) and set this bit to 1. The Command Timeout Error shall also be set to 1 to distinguish ***CMD*** line conflict. '1' = CRC Error generated '0' = No Error	0
STACMDTOUTERR	[0]	**Command Timeout Error** Occurs only if no response is returned within 64 ***SDCLK*** cycles from the end bit of the command. If the Host Controller detects a ***CMD*** line conflict, in which case **Command CRC Error shall** also be set as shown in Table 33, this bit **shall** be set without waiting for 64 ***SDCLK*** cycles because the command will be aborted by the Host Controller. '1' = Timeout '0' = No Error	0

The relation between **Command CRC Error** and **Command Timeout Error** is shown in Table below.

The relation between Command CRC Error and Command Timeout Error

Command CRC Error	Command Timeout Error	Kinds of error
0	0	No Error
0	1	Response Timeout Error
1	0	Response CRC Error
1	1	***CMD*** line conflict

27.7.20 NORMAL INTERRUPT STATUS ENABLE REGISTER

Setting to 1 enables Interrupt Status.

Register	Address	R/W	Description	Reset Value
NORINTSTSEN0	0x7C200034	R/W	Normal Interrupt Status Enable Register (Channel 0)	0x0
NORINTSTSEN1	0x7C300034	R/W	Normal Interrupt Status Enable Register (Channel 1)	0x0
NORINTSTSEN2	0x7C400034	R/W	Normal Interrupt Status Enable Register (Channel 2)	0x0

Name	Bit	Description	Initial Value
	[15]	**Fixed to 0** The Host Driver shall control error interrupts using the *Error Interrupt Status Enable* register. *(RO)*	0
ENSTAFIA3	[14]	**FIFO SD Address Pointer Interrupt 3 Status Enable** '1' = Enabled '0' = Masked	0
ENSTAFIA2	[13]	**FIFO SD Address Pointer Interrupt 2 Status Enable** '1' = Enabled '0' = Masked	0
ENSTAFIA1	[12]	**FIFO SD Address Pointer Interrupt 1 Status Enable** '1' = Enabled '0' = Masked	0
ENSTAFIA0	[11]	**FIFO SD Address Pointer Interrupt 0 Status Enable** '1' = Enabled '0' = Masked	0
ENSTARWAIT	[10]	**Read Wait interrupt status enable** '1' = Enabled '0' = Masked	0
ENSTACCS	[9]	**CCS Interrupt Status Enable** '1' = Enabled '0' = Masked	0
ENSTACARDINT	[8]	**Card Interrupt Status Enable** If this bit is set to 0, the Host Controller clears interrupt request to the System. The **Card Interrupt** detection is stopped when this bit is cleared and restarted when this bit is set to 1. The Host Driver must clear the **Card Interrupt Status Enable** before servicing the **Card Interrupt** and must set this bit again after all interrupt requests from the card are cleared to prevent inadvertent interrupts. '1' = Enabled '0' = Masked	0

Name	Bit	Description	Initial Value
ENSTACARDREM	[7]	**Card Removal Status Enable** '1' = Enabled '0' = Masked	0
ENSTACARDNS	[6]	**Card Insertion Status Enable** '1' = Enabled '0' = Masked	0
ENSTABUFRDRDY	[5]	**Buffer Read Ready Status Enable** '1' = Enabled '0' = Masked	0
ENSTABUFWTRDY	[4]	**Buffer Write Ready Status Enable** '1' = Enabled '0' = Masked	0
ENSTADMA	[3]	**DMA Interrupt Status Enable** '1' = Enabled '0' = Masked	0
ENSTABLKGAP	[2]	**Block Gap Event Status Enable** '1' = Enabled '0' = Masked	0
ENSTASTANSCMPLT	[1]	**Transfer Complete Status Enable** '1' = Enabled '0' = Masked	0
ENSTACMDCMPLT	[0]	**Command Complete Status Enable** '1' = Enabled '0' = Masked	0

27.7.21 ERROR INTERRUPT STATUS ENABLE REGISTER

Setting to 1 enables Error Interrupt Status.

Register	Address	R/W	Description	Reset Value
ERRINTSTSEN0	0x7C200036	R/W	Error Interrupt Status Enable Register (Channel 0)	0x0
ERRINTSTSEN1	0x7C300036	R/W	Error Interrupt Status Enable Register (Channel 1)	0x0
ERRINTSTSEN2	0x7C400036	R/W	Error Interrupt Status Enable Register (Channel 2)	0x0

Name	Bit	Description	Initial Value
	[15:9]	Reserved	0
ENSTAACMDERR	[8]	**Auto CMD12 Error Status Enable** '1' = Enabled '0' = Masked	0
ENSTACURERR	[7]	**Current Limit Error Status Enable** This function is not implemented in this version. '1' = Enabled '0' = Masked	0
ENSTADENDERR	[6]	**Data End Bit Error Status Enable** '1' = Enabled '0' = Masked	0
ENSTADATCRCERR	[5]	**Data CRC Error Status Enable** '1' = Enabled '0' = Masked	0
ENSTADATTOUTERR	[4]	**Data Timeout Error Status Enable** '1' = Enabled '0V = Masked	0
ENSTACMDIDXERR	[3]	**Command Index Error Status Enable** '1' = Enabled '0' = Masked	0
ENSTACMDEBITERR	[2]	**Command End Bit Error Status Enable** '1' = Enabled '0' = Masked	0
ENSTACMDCRCERR	[1]	**Command CRC Error Status Enable** '1' = Enabled '0' = Masked	0
ENSTACMDTOUTERR	[0]	**Command Timeout Error Status Enable** '1' = Enabled '0' = Masked	0

27.7.22 NORMAL INTERRUPT SIGNAL ENABLE REGISTER

This register is used to select which interrupt status is indicated to the Host System as the interrupt. These status bits all share the same1 bit interrupt line. To enable interrupt generate set any of this bit to 1.

Register	Address	R/W	Description	Reset Value
NORINTSIGEN0	0x7C200038	R/W	Normal Interrupt Signal Enable Register (Channel 0)	0x0
NORINTSIGEN1	0x7C300038	R/W	Normal Interrupt Signal Enable Register (Channel 1)	0x0
NORINTSIGEN2	0x7C400038	R/W	Normal Interrupt Signal Enable Register (Channel 2)	0x0

Name	Bit	Description	Initial Value
	[15]	**Fixed to 0** The Host Driver shall control error interrupts using the *Error Interrupt Signal Enable* register.	0
ENSIGFIA3	[14]	**FIFO SD Address Pointer Interrupt 3 Signal Enable** '1' = Enabled '0' = Masked	0
ENSIGFIA2	[13]	**FIFO SD Address Pointer Interrupt 2 Signal Enable** '1' = Enabled V0' = Masked	0
ENSIGFIA1	[12]	**FIFO SD Address Pointer Interrupt 1 Signal Enable** '1' = Enabled '0' = Masked	0
ENSIGFIA0	[11]	**FIFO SD Address Pointer Interrupt 0 Signal Enable** '1' = Enabled '0' = Masked	0
ENSIGRWAIT	[10]	**Read Wait Interrupt Signal Enable** '1' = Enabled '0' = Masked	0
ENSIGCCS	[9]	**CCS Interrupt Signal Enable** Command Complete Singal Interrupt Status bit is for CE-ATA interface mode. '1' = Enabled '0' = Masked	0
ENSIGCARDINT	[8]	**Card Interrupt Signal Enable** '1' = Enabled '0' = Masked	0

Name	Bit	Description	Initial Value
ENSIGCARDREM	[7]	**Card Removal Signal Enable** '1' = Enabled '0' = Masked	0
ENSIGCARDNS	[6]	**Card Insertion Signal Enable** '1' = Enabled '0' = Masked	0
ENSIGBUFRDRDY	[5]	**Buffer Read Ready Signal Enable** '1' = Enabled '0' = Masked	0
ENSIGBUFWTRDY	[4]	**Buffer Write Ready Signal Enable** '1' = Enabled '0' = Masked	0
ENSIGDMA	[3]	**DMA Interrupt Signal Enable** '1' = Enabled '0' = Masked	0
ENSIGBLKGAP	[2]	**Block Gap Event Signal Enable** '1' = Enabled '0' = Masked	0
ENSIGSTANSCMPLT	[1]	**Transfer Complete Signal Enable** '1' = Enabled '0' = Masked	0
ENSIGCMDCMPLT	[0]	**Command Complete Signal Enable** '1' = Enabled '0' = Masked	0

27.7.23 ERROR INTERRUPT SIGNAL ENABLE REGISTER

This register is used to select which interrupt status is notified to the Host System as the interrupt. These status bits all share the same 1 bit interrupt line. To enable interrupt generate set any of this bit to 1.

Register	Address	R/W	Description	Reset Value
ERRINTSIGEN0	0x7C20003A	R/W	Error Interrupt Signal Enable Register (Channel 0)	0x0
ERRINTSIGEN1	0x7C30003A	R/W	Error Interrupt Signal Enable Register (Channel 1)	0x0
ERRINTSIGEN2	0x7C40003A	R/W	Error Interrupt Signal Enable Register (Channel 2)	0x0

Name	Bit	Description	Initial Value
	[15:9]	Reserved	0
ENSIGACMDERR	[8]	**Auto CMD12 Error Signal Enable** '1' = Enabled '0' = Masked	0
ENSIGCURERR	[7]	**Current Limit Error Signal Enable** This function is not implemented in this version. '1' = Enabled '0' = Masked	0
ENSIGDENDERR	[6]	**Data End Bit Error Signal Enable** '1' = Enabled '0' = Masked	0
ENSIGDATCRCERR	[5]	**Data CRC Error Signal Enable** '1' = Enabled '0' = Masked	0
ENSIGDATTOUTERR	[4]	**Data Timeout Error Signal Enable** '1' = Enabled '0' = Masked	0
ENSIGCMDIDXERR	[3]	**Command Index Error Signal Enable** '1' = Enabled '0' = Masked	0
ENSIGCMDEBITERR	[2]	**Command End Bit Error Signal Enable** '1' = Enabled '0' = Masked	0
ENSIGCMDCRCERR	[1]	**Command CRC Error Signal Enable** '1' = Enabled '0' = Masked	0
ENSIGCMDTOUTERR	[0]	**Command Timeout Error Signal Enable** '1' = Enabled	0
		'0' = Masked	

Detailed documents are to be copied from SD Host Standard Specification.

27.7.24 AUTOCMD12 ERROR STATUS REGISTER

When *Auto CMD12 Error Status* is set, the Host Driver checks this register to identify what kind of error Auto CMD12 indicated. This register is valid only when the **Auto CMD12 Error** is set.

Register	Address	R/W	Description	Reset Value
ACMD12ERRSTS0	0x7C20003C	ROC	Auto CMD12 Error Status Register (Channel 0)	0x0
ACMD12ERRSTS1	0x7C30003C	ROC	Auto CMD12 Error Status Register (Channel 1)	0x0
ACMD12ERRSTS2	0x7C40003C	ROC	Auto CMD12 Error Status Register (Channel 2)	0x0

Name	Bit	Description	Initial Value
	[15:8]	Reserved	0
STANCMDAER	[7]	**Command Not Issued By Auto CMD12 Error** Setting this bit to 1 means CMD_wo_DAT is not executed due to an Auto CMD12 Error (D04-D01) in this register. '1' = Not Issued '0' = No error	0
	[6:5]	**Reserved**	0
STACMDIDXERR	[4]	**Auto CMD12 Index Error** Occurs if the Command Index error occurs in response to a command. '1' = Error '0' = No Error	0
STACMDEBITAER	[3]	**Auto CMD12 End Bit Error** Occurs when detecting that the end bit of command response is 0. '1' = End Bit Error Generated '0' = No Error	0
STACMDCRCAER	[2]	**Auto CMD12 CRC Error** Occurs when detecting a CRC error in the command response. '1' = CRC Error Generated '0' = No Error	0
STACMDTOUTAER	[1]	**Auto CMD12 Timeout Error** Occurs if no response is returned within 64 ***SDCLK*** cycles from the end bit of command. If this bit is set to1, the other error status bits (D04-D02) are meaningless. '1' = Time out '0' = No Error	0
STANACMDAER	[0]	**Auto CMD12 Not Executed** If memory multiple block data transfer is not started due to command error, this bit is not set because it is not necessary to issue Auto CMD12. Setting this bit to 1 means the Host Controller cannot issue Auto CMD12 to stop memory multiple block data transfer due to some error. If this bit is	0
		set to 1, other error status bits (D04-D01) are meaningless. '1' = Not executed '0' = Executed	

The relation between Auto CMD12 CRC Error and Auto CMD12 Timeout Error is shown below.

The relation between Command CRC Error and Command Timeout Error

Auto CMD12 CRC Error	Auto CMD12 Timeout Error	Kinds of error
0	0	No Error
0	1	Response Timeout Error
1	0	Response CRC Error
1	1	***CMD*** line conflict

The timing of changing *Auto CMD12 Error Status* can be classified in three scenarios:

(1) When the Host Controller is going to issue Auto CMD12

Set D00 to 1 if Auto CMD12 cannot be issued due to an error in the previous command.

Set D00 to 0 if Auto CMD12 is issued.

(2) At the end bit of an Auto CMD12 response

Check received responses by checking the error bits D01, D02, D03 and D04.

Set to 1 if error is detected.

Set to 0 if error is not detected.

(3) Before reading the Auto CMD12 Error Status bit D07

Set D07 to 1 if there is a command cannot be issued

Set D07 to 0 if there is no command to issue

Timing of generating the **Auto CMD12 Error** and writing to the *Command* register are asynchronous. Then D07 are sampled when driver never writing to the *Command* register. So just before reading the *Auto CMD12 Error Status* register set the D07 status bit. An Auto CMD12 Error Interrupt is generated when one of the error bits D00 to D04 is set to 1. The **Command Not Issued By Auto CMD12 Error** does not generate an interrupt.

27.7.25 CAPABILITIES REGISTER

This register provides the Host Driver with information specific to the Host Controller implementation. The Host Controller may implement these values as fixed or loaded from flash memory during power on initialization. Refer to **Software Reset For All** in the *Software Reset* register for loading from flash memory and completion timing control.

Register	Address	R/W	Description	Reset Value
CAPAREG0	0x7C200040	HWInit	Capabilities Register (Channel 0)	0x05E00080
CAPAREG1	0x7C300040	HWInit	Capabilities Register (Channel 1)	0x05E00080
CAPAREG2	0x7C400040	HWInit	Capabilities Register (Channel 2)	0x05E00080

Name	Bit	Description	Initial Value
	[31:27]	Reserved	
CAPAV18	[26]	**Voltage Support 1.8V** (HWInit) '1' = 1.8V Supported '0' = 1.8V Not Supported	1
CAPAV30	[25]	**Voltage Support 3.0V** (HWInit) '1' = 3.0V Supported '0' = 3.0V Not Supported	0
CAPAV33	[24]	**Voltage Support 3.3V** (HWInit) '1' = 3.3V Supported '0' = 3.3V Not Supported	1
CAPASUSRES	[23]	**Suspend/Resume Support** (HWInit) This bit indicates whether the Host Controller supports Suspend / Resume functionality. If this bit is 0, the Suspend and Resume mechanism are not supported and the Host Driver does not issue either Suspend or Resume commands. '1' = Supported '0' = Not Supported	1
CAPADMA	[22]	**DMA Support** (HWInit) This bit indicates whether the Host Controller is capable of using DMA to transfer data between system memory and the Host Controller directly. '1' = DMA Supported '0' = DMA Not Supported	1
CAPAHSPD	[21]	**High Speed Support** (HWInit) This bit indicates whether the Host Controller and the Host System support High Speed mode and they can supply SD Clock frequency from 25MHz to 50MHz. '1' = High Speed Supported '0' = High Speed Not Supported	1
	[20:18]	**Reserved**	0

Name	Bit	Description	Initial Value
CAPAMAXBLKLEN	[17:16]	**Max Block Length** (HWInit) This value indicates the maximum block size that the Host Driver can read and write to the buffer in the Host Controller. The buffer transfers this block size without wait cycles. Three sizes can be defined as indicated below. '00'=512-byte, '01'=1024-byte, '10'=2048-byte, '11'=Reserved	0
	[15:14]	**Reserved**	0
CAPABASECLK	[13:8]	**Base Clock Frequency For SD Clock** (HWInit) This value indicates the base (maximum) clock frequency for the SD Clock. Unit values are 1MHz. If the real frequency is 16.5MHz, the lager value is set to 01 0001b (17MHz) because the Host Driver use this value to calculate the clock divider value (Refer to the **SDCLK Frequency Select** in the *Clock Control* register.) and it does not exceed upper limit of the SD Clock frequency. The supported clock range is 10MHz to 63MHz. If these bits are all 0, the Host System has to get information via another method. Not '0'=1MHz to 63MHz 000000b = Get information via another method	0
CAPATOUTUNIT	[7]	**Timeout Clock Unit** (HWInit) This bit shows the unit of base clock frequency used to detect **Data Timeout Error**. '0' = kHz, '1' = MHz	1
	[6]	**Reserved**	0
CAPATOUTCLK	[5:0]	**Timeout Clock Frequency** (HWInit) This bit shows the base clock frequency used to detect **Data Timeout Error**. The **Timeout Clock Unit** defines the unit of this field value. **Timeout Clock Unit** =0 [kHz] unit: 1kHz to 63kHz **Timeout Clock Unit** =1 [MHz] unit: 1MHz to 63MHz Not 0 = 1kHz to 63kHz or 1MHz to 63MHz 00 0000b = Get information via another method	0

27.7.26 MAXIMUM CURRENT CAPABILITIES REGISTER

These registers indicate maximum current capability for each voltage. The value is meaningful if **Voltage Support** is set in the *Capabilities* register. If this information is supplied by the Host System via another method, all *Maximum Current Capabilities* register will be 0.

Register	Address	R/W	Description	Reset Value
MAXCURR0	0x7C200048	HWInit	Maximum Current Capabilities Register (Channel 0)	0x0
MAXCURR1	0x7C300048	HWInit	Maximum Current Capabilities Register (Channel 1)	0x0
MAXCURR2	0x7C400048	HWInit	Maximum Current Capabilities Register (Channel 2)	0x0

Name	Bit	Description	Initial Value
	[31:24]	Reserved	
MAXCURR18	[23:16]	**Maximum Current for 1.8V** (HWInit)	0
MAXCURR30	[15:8]	**Maximum Current for 3.0V** (HWInit)	0
MAXCURR33	[7:0]	**Maximum Current for 3.3V** (HWInit)	0

This register measures current in 4mA steps. Each voltage level's current support is described using the Table below.

Maximum Current Value Definition

Register Value	Current Value
0	Get information via another method
1	4mA
2	8mA
3	12mA
…	…
255	1020mA

27.7.27 CONTROL REGISTER 2

This register contains the SD Command Argument.

Register	Address	R/W	Description	Reset Value
CONTROL2_0	0x7C200080	R/W	Control register 2 (Channel 0)	0x0
CONTROL2_1	0x7C300080	R/W	Control register 2 (Channel 1)	0x0
CONTROL2_2	0x7C400080	R/W	Control register 2 (Channel 2)	0x0

Name	Bit	Description	Initial Value
-	[31]	Reserved	0
ENCMDCNFMSK	[30]	**Command Conflict Mask Enable** This bit can mask enable the Command Conflict Status (bit [1:0] of the "ERROR INTERRUPT STATUS REGISTER") 0=Mask Disable, 1=Mask Enable Note) When the ENHIGHSPD field in the Host Control Register is set (High Speed data transfer), this field should be enabled to prevent from command conflict status alarm.	0
CDINVRXD3	[29]	**Card Detect signal inversion for RX_DAT[3]** 0=Disable, 1=Enable	0
SELCARDOUT	[28]	**Card Removed Condition Selection** 0= Card Removed condition is "Not Card Insert" State (When the transition from "Card Inserted" state to "Debouncing" state in Figure 27-13) 1= Card Removed state is "Card Out" State (When the transition from Debouncing state to "No Card" state in Figure 27-13)	0
FLTCLKSEL	[27:24]	**Filter Clock (iFLTCLK) Selection** Filter Clock period = 2^(FltClkSel + 5) x iSDCLK period 0000 = 25 x iSDCLK, 0001 = 26 x iSDCLK … 1111 = 220 x iSDCLK	0
LVLDAT	[23:16]	**DAT line level** Bit[23]=DAT[7], BIT[22]=DAT[6], BIT[21]=DAT[5], BIT[20]=DAT[4], Bit[19]=DAT[3], BIT[18]=DAT[2], BIT[17]=DAT[1], BIT[16]=DAT[0] (Read Only)	Line state
ENFBCLKTX	[15]	**Feedback Clock Enable for Tx Data/Command Clock** '0'=Disable, '1'=Enable	0
ENFBCLKRX	[14]	**Feedback Clock Enable for Rx Data/Command Clock** '0'=Disable, '1'=Enable	0
SDCDSEL	[13]	**SD Card Detect Signal Selection** Card Detect Pin Level does not simply reflect SDCD# pin, but chooses from SDCD, DAT[3], or CDTestlvl depending on CDSigSel and this field (SDCDSel) values '0'=nSDCD is used for SD Card Detect Signal '1'=DAT[3] is used for SD Card Detect Signal	0

Name	Bit	Description	Initial Value
SDSIGPC	[12]	**SD Output Signal Power Control Support** This field is used to enable output CMD and DAT referencing SD Bus Power bit in the "PWRCON register", when being set. '0'= CMD and DAT outputs are not controlled by SD Bus Power bit '1'= CMD and DAT outputs are controlled(masked) by SD Bus Power bit	0
ENBUSYCHKTXSTART	[11]	**CE-ATA I/F mode** Busy state check before Tx Data start state 0=Disable, 1=Enable	0
DFCNT	[10:9]	**Debounce Filter Count** Debounce Filter Count setting register for Card Detect signal input (SDCD#) 00=No use debounce filter, 01=4 iSDCLK, 10=16 iSDCLK, 11=64 iSDCLK	0
ENCLKOUTHOLD	[8]	**SDCLK Hold Enable** The enter and exit of the SDCLK Hold state is done by Host Controller. 0=Disable, 1=Enable	0
RWAITMODE	[7]	**Read Wait Release Control** 0=Read Wait state is released by the Host Controller (Auto) 1=Read Wait state is released by the Host Driver (Manual)	0
DISBUFRD	[6]	**Buffer Read Disable** 0=Normal mode, user can read buffer(FIFO) data using 0x20 register 1=User cannot read buffer(FIFO) data using 0x20 register. In this case, the buffer memory only can be read through memory area. (Debug purpose)	0
SELBASECLK	[5:4]	**Base Clock Source Select** 00 or 01 =HCLK, 10=EPLL out Clock (from SYSCON), 11=External Clock source (XTI or XEXTCLK)	00
PWRSYNC	[3]	**SD OP Power Sync Support with SD Card** This field is used to enable input CMD and DAT referencing SD Bus Power bit in the "PWRCON register", when being set. '0'=No Sync, no switch input enable signal (Command, Data) '1'=Sync, control input enable signal (Command, Data)	0
-	[2]	**Reserved**	0
ENCLKOUTMSKCON	[1]	**SDCLK output clock masking when Card Insert cleared** This field when High is used not to stop SDCLK when No Card state. '0'=Disable, '1'=Enable	0
HWINITFIN	[0]	**SD Host Controller Hardware Initialization Finish** 0=Not Finish, 1=Finish	0

NOTES:

1. Ensure to always set SDCLK Hold Enable (EnSCHold) if the card does not support Read Wait to guarantee for Receive data not overwritten to the internal FIFO memory.
2. CMD_wo_DAT issue is prohibited during READ transfer when SDCLK Hold Enable is set.

27.2.28 CONTROL REGISTERS 3 REGISTER

Register	Address	R/W	Description	Reset Value
CONTROL3_0	0x7C200084	R/W	FIFO Interrupt Control (Control Register 3) (Channel 0)	0x7F5F3F1F
CONTROL3_1	0x7C300084	R/W	FIFO Interrupt Control (Control Register 3) (Channel 1)	0x7F5F3F1F
CONTROL3_2	0x7C400084	R/W	FIFO Interrupt Control (Control Register 3) (Channel 2)	0x7F5F3F1F

Name	Bit	Description	Initial Value
FCSEL3	[31]	**Feedback Clock Select [3]** Reference Note (1)	0x0
FIA3	[30:24]	**FIFO Interrupt Address register 3** FIFO (512Byte Buffer memory, word address unit) Initial value(0x7F) generates at 512-byte(128-word) position.	0x7F
FCSEL2	[23]	**Feedback Clock Select [2]** Reference Note (1)	0x0
FIA2	[22:16]	**FIFO Interrupt Address register 2** FIFO (512Byte Buffer memory, word address unit) Initial value(0x5F) generates at 384-byte(96-word) position.	0x5F
FCSEL1	[15]	**Feedback Clock Select [1]** Reference Note (2)	0x0
FIA1	[14:8]	**FIFO Interrupt Address register 1** FIFO (512Byte Buffer memory, word address unit) Initial value(0x3F) generates at 256-byte(64-word) position.	0x3F
FCSEL0	[7]	**Feedback Clock Select [0]** Reference Note (2)	0x0
FIA0	[6:0]	**FIFO Interrupt Address register 0** FIFO (512Byte Buffer memory, word address unit) Initial value(0x1F) generates at 128-byte(32-word) position.	0x1F

NOTES:

1. FCSel[3:2] : Tx Feedback Clock Delay Control : Inverter delay means 10ns delay when SDCLK 50MHz setting '01'=Delay1 (basic delay), '11'=Delay2 (basic delay + 2ns), '00'=Delay3 (inverter delay), '10'=Delay4 (inverter delay + 2ns)
2. FCSel[1:0] : Rx Feedback Clock Delay Control : Inverter delay means10ns delay when SDCLK 50MHz setting : '00' or '01' = Delay1 (inverter delay), '10' or '11' = Delay2 (inverter delay + 2ns)
3. Tx Feedback inversion setting (FCSel[3:2] = '00' or '10') and Normal Speed mode (ENHIGHSPD = 0) setting make Tx data transfer mismatch (Do not set).

27.7.29 HOST CONTROLLER VERSION REGISTER

This register contains the SD Command Argument.

Register	Address	R/W	Description	Reset Value
HCVER0	0x7C2000FE	HWInit	Host Controller Version Register (Channel 0)	0x1300
HCVER1	0x7C3000FE	HWInit	Host Controller Version Register (Channel 1)	0x1300
HCVER2	0x7C4000FE	HWInit	Host Controller Version Register (Channel 2)	0x1300

Name	Bit	Description	Initial Value
VENVER	[15:8]	**Vendor Version Number** This status is reserved for the vendor version number. The Host Driver must not use this status. 0x13 : SDMMC3.1 Host Controller	0x13
SPECVER	[7:0]	**Specification Version Number** This status indicates the Host Controller Spec. Version. The upper and lower 4-bit indicate the version. '00' = SD Host Specification Version 1.0 Ohters = Reserved	0x00

Revision History

Date	Editor	Version	Descriptions
2008-08-12	Sejong Lee	1.0	최초 작성