TRAVEO™ T2G ファミリ MCU の SDHC ホストコントローラの使用方法

本書について

適用範囲と目的

AN224414 は TRAVEO™ T2G ファミリ MCU に搭載されている Secure Digital High Capacity (SDHC) ホストコントローラを使用したデータ転送シーケンスについて説明します。

対象者

本書は、TRAVEO™ T2G ファミリの SDHC ホストコントローラを使用するユーザーを対象とします。

関連製品ファミリ

TRAVEO™ T2G ファミリ CYT3/CYT4/CYT6 シリーズ

目次

本書について **1**
目次 **2**
1 はじめに **3**
2 データ転送シーケンス **4**
2.1 DMA 未使用時のデータ転送シーケンス 5
2.2 シングルオペレーション DMA (SDMA) 使用時のデータ転送シーケンス 7
2.3 アドバンスド DMA (ADMA) 使用時のデータ転送シーケンス 9
3 アボート転送シーケンス **11**
3.1 アボートコマンドシーケンス 11
3.2 非同期アボートシーケンス 12
3.3 同期アボートシーケンス 12
3.4 リセットコマンド 13
関連ドキュメント **14**
用語集 **15**
改訂履歴 **16**
免責事項 **17**

1 はじめに

このアプリケーションノートはインフィニオン TRAVEO™ T2G ファミリの SDHC ホストコントローラを使用したデータとアボート転送シーケンスについて説明します。SDHC ホストコントローラは、Embedded Multimedia Card (eMMC) ベースのメモリデバイス, secure Digital (SD) カードおよび Secure Digital Input Output (SDIO) カードとのインタフェースを可能にします。このアプリケーションノートは、direct memory access (DMA) 使用時および未使用時の転送のシーケンスを説明します。初期化シーケンスについては、architecture reference manual の SDHC Host Controller の章を参照してください。

このアプリケーションノートは、CYT3/4/6 シリーズの MCU を対象とします。CYT3 シリーズは、1 つの Arm® Cortex®-M7 ベースの CPU (CM7) と 1 つの Cortex-M0+ベースの CPU (CM0+) を持ち、CYT4 シリーズは、2 つの CM7 CPU と 1 つの CM0+ CPU を持ち、CYT6 シリーズは、4 つの CM7 CPU と 1 つの CM0+ CPU を持ちます。

このアプリケーションノートで使用されている機能と用語を理解するために architecture reference manual の SDHC Host Controller の章を参照してください。

2 データ転送シーケンス

DMA の使用の有無によって、2 つの実行方法があります。DMA 未使用時のシーケンスを Figure 1 に示します。DMA 使用時のシーケンスを Figure 2 および Figure 3 に示します。

また転送シーケンスは、ブロック数の設定に応じて、以下のタイプに分類されます。

- シングルブロック転送: ブロック数は、転送前にホストコントローラに設定されます。ブロック数は常に 1 です。
- マルチブロック転送: ブロック数は、転送前にホストコントローラに設定されます。設定されたブロック数は 1 以上です。
- 無限ブロック転送: ブロック数は、転送前にホストコントローラに設定されません。この転送は、アボート転送が実行されるまで続きます。このアボート転送は、SD メモリカードと eMMC の場合は CMD12、SDIO カードの場合は CMD52 によって実行されます。

2 データ転送シーケンス

2.1 DMA 未使用時のデータ転送シーケンス

DMA 未使用時のデータ転送シーケンスを Figure 1 に示します。

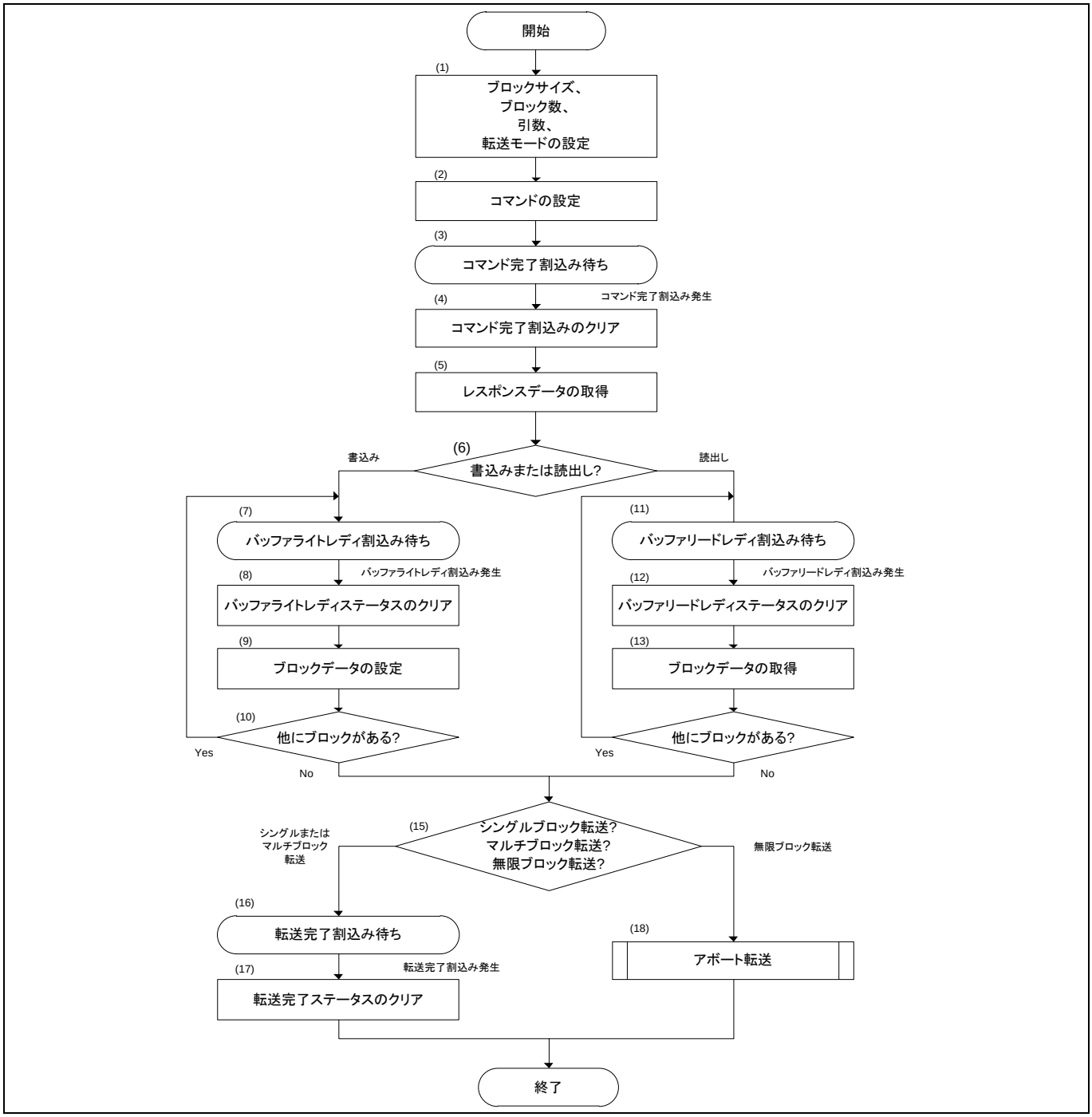

Figure 1 DMA 未使用時のデータ転送シーケンス

2 データ転送シーケンス

上記のフローチャートで番号表示されている箇所を以下に説明します。

1. 以下の設定をします。
   a) 必要なブロックの長さをブロックサイズレジスタ (SDHC_CORE_BLOCKSIZE_R) に設定します。
   b) 必要なデータブロック数をブロックカウントレジスタ (SDHC_CORE_BLOCKCOUNT_R) に設定します。
   c) コマンドフォーマットのビット 8〜39 で指定されているコマンド引数を引数レジスタ (SDHC_CORE_ARGUMENT_R) に設定します。
   d) 転送モードレジスタ (SDHC_CORE_XFER_MODE_R) を設定します。ホストドライバは、マルチ/シングルブロック選択, ブロックカウント有効, データ転送方向, auto CMD12 有効, および DMA 有効を決定します。レスポンス確認が有効 (RESP_ERR_CHK_ENABLE = '1') な場合、レスポンス割込みディセーブル (RES_INT_DISABLE) を'1'に設定します。そして、レスポンスタイプ (RESP_TYPE) は、R1 (0: メモリ) または R5 (1: SDIO) を選択します。
2. 実行するコマンドをコマンドレジスタ (SDHC_CORE_CMD_R) に設定します。
3. レスポンス確認を有効にしている場合は、ステップ 6 に進みます。そうではない場合は、コマンド完了割込みを待ちます。
4. ノーマル割込みステータスレジスタのコマンド完了ビット (SDHC_CORE_NORMAL_INT_STAT_R.CMD_COMPLETE) をクリアするために、'1'を書き込みます。
5. 発行されたコマンドに対する必要なレスポンスを取得するためにレスポンスレジスタ (SDHC_CORE_RESPXX_R) を読み出します。
6. カードに書き込む場合、ステップ 7 に進みます。カードから読み出す場合、ステップ 11 に進みます。
7. バッファライトレディ割込みを待ちます。
8. ノーマル割込みステータスレジスタのバッファライトレディビット (SDHC_CORE_NORMAL_INT_STAT_R.BUF_WR_READY) をクリアするために、'1'を書き込みます。
9. バッファデータポートレジスタ (SDHC_CORE_BUF_DATA_R) に (ステップ 1 で指定したバイト数に従って) ブロックデータを書き込みます。
10. すべてのブロックが送信されるまで、ステップ 7 から 9 を繰り返して、ステップ 15 に進みます。
11. バッファリードレディ割込みを待ちます。
12. ノーマル割込みステータスレジスタのバッファリードレディビット (SDHC_CORE_NORMAL_INT_STAT_R.BUF_RD_READY) をクリアするために、'1'を書き込みます。
13. バッファデータポートレジスタ (SDHC_CORE_BUF_DATA_R) から (ステップ 1 で指定したバイト数に従って) ブロックデータを読み出します。
14. すべてのブロックを受信するまで、ステップ 11 から 13 を繰り返して、ステップ 15 に進みます。
15. このシーケンスがシングルまたはマルチブロック転送の場合は、ステップ 16 に進みます。無限ブロック転送の場合は、ステップ 18 に進みます。
16. 転送完了割込みを待ちます。
17. ノーマル割込みステータスレジスタの転送完了ビット (SDHC_CORE_NORMAL_INT_STAT_R.XFER_COMPLETE) をクリアするために、'1'を書き込みます。
18. アボート転送シーケンスに従って、アボート転送のシーケンスを実行します。

2.2 シングルオペレーション DMA (SDMA) 使用時のデータ転送シーケンス

SDMA 使用時のデータ転送シーケンスを Figure 2 に示します。

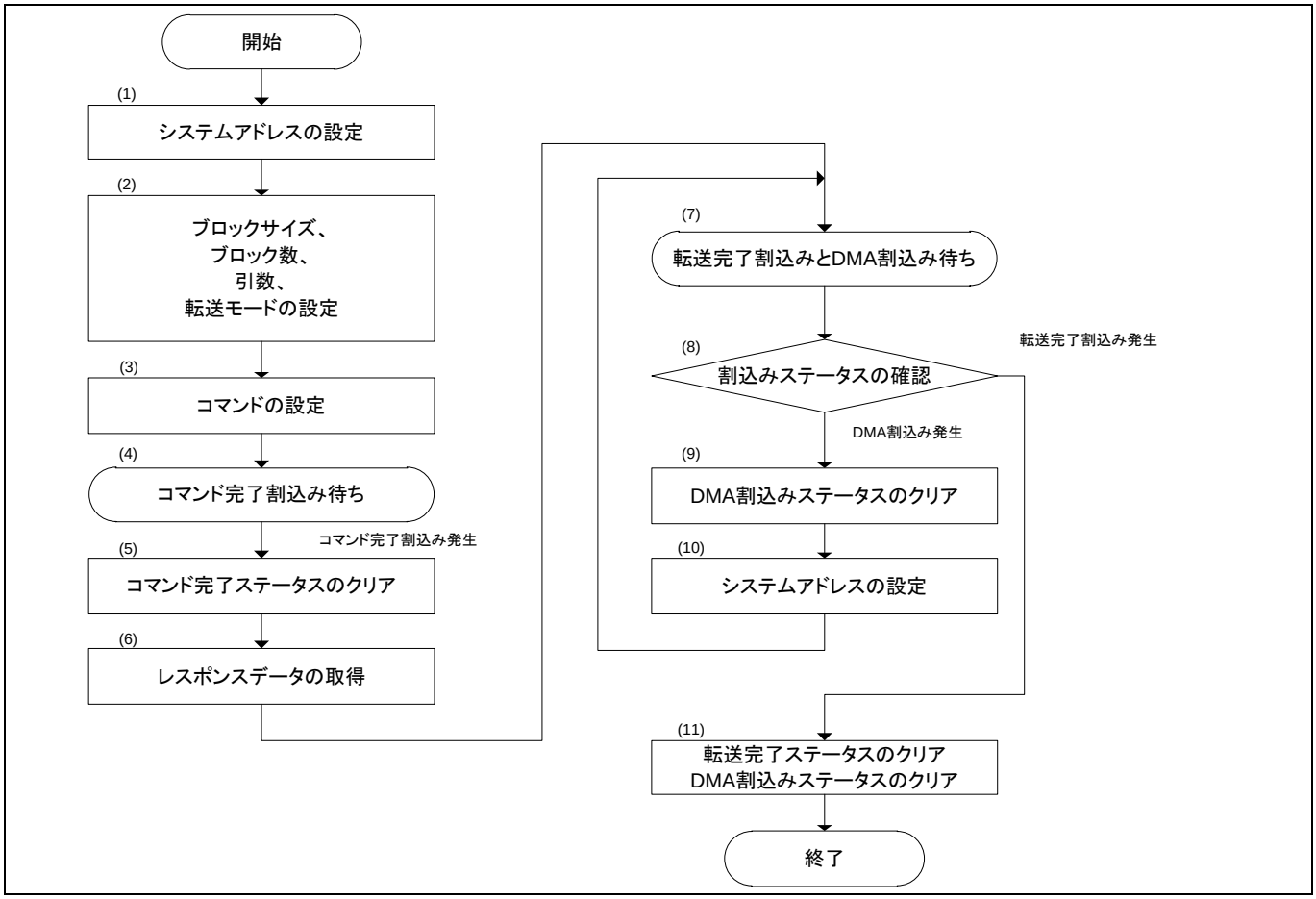

Figure 2 シングルオペレーション DMA (SDMA) 使用時のデータ転送シーケンス

上記のフローチャートで番号表示されている箇所を以下に説明します。

1. システムメモリのデータの場所を、ホストバージョン 4 イネーブル (SDHC_CORE_HOST_CTRL2_R.HOST_VER4_ENABLE) = '0'の場合、SDMA システムアドレスレジスタ (SDHC_CORE_SDMASA_R) に設定し、ホストバージョン 4 イネーブル (SDHC_CORE_HOST_CTRL2_R.HOST_VER4_ENABLE) = '1'の場合、ADMA システムアドレスレジスタ (SDHC_CORE_ADMA_SA_LOW_R) に設定します。
2. 以下の設定をします
   a) 必要なブロックの長さをブロックサイズレジスタ (SDHC_CORE_BLOCKSIZE_R) に設定します。
   b) 必要なデータブロック数をブロックカウントレジスタに設定します。
   SDHC_CORE_HOST_CTRL2_R.HOST_VER4_ENABLE = '0'または 16 ビットブロックカウントレジスタ (SDHC_CORE_BLOCKCOUNT_R) に 0 以外が設定されている場合、16 ビットブロックカウントレジスタが選択されます。SDHC_CORE_HOST_CTRL2_R.HOST_VER4_ENABLE = '1'かつ 16 ビットブロックカウントレジスタに 0x0000 が設定されている場合、32 ビットブロックカウントレジスタ (SDHC_CORE_SDMASA_R) が選択されます。
   c) コマンドフォーマットのビット 8～39 で指定されているコマンド引数を引数レジスタ (SDHC_CORE_ARGUMENT_R) に設定します。

2 データ転送シーケンス

d) 転送モードレジスタ (SDHC_CORE_XFER_MODE_R) を設定します。ホストドライバは、マルチ/シングルブロック選択, ブロックカウント有効, データ転送方向, Auto CMD12 有効, および DMA 有効を決定します。レスポンス確認が有効 (RESP_ERR_CHK_ENABLE = '1') な場合、レスポンス割込みディセーブル (RES_INT_DISABLE) を'1'に設定します。そして、レスポンスタイプ (RESP_TYPE) は、R1 (0: メモリ) または R5 (1: SDIO) を選択します。

3. 実行するコマンドをコマンドレジスタ (SDHC_CORE_CMD_R) に設定します。
4. レスポンス確認を有効にしている場合は、ステップ 7 に進みます。そうではない場合は、コマンド完了割込みを待ちます。
5. ノーマル割込みステータスレジスタのコマンド完了ビット (SDHC_CORE_NORMAL_INT_STAT_R.CMD_COMPLETE) をクリアするために、'1'を書き込みます。
6. 発行されたコマンドに対する必要なレスポンスを取得するためにレスポンスレジスタ (SDHC_CORE_RESPXX_R) を読み出します。
7. データ転送完了割込み (SDHC_CORE_NORMAL_INT_STAT_R.XFER_COMPLETE) と DMA 割込み (SDHC_CORE_NORMAL_INT_STAT_R.DMA_INTERRUPT) を待ちます。
8. データ転送完了ビット (SDHC_CORE_NORMAL_INT_STAT_R.XFER_COMPLETE) が'1'の場合、ステップ 11 に進みます。そうではなく DMA 割込みビット (SDHC_CORE_NORMAL_INT_STAT_R.DMA_INTERRUPT) が'1'の場合、ステップ 9 に進みます。データ転送完了割込みは、DMA 割込みよりも優先度が高い割込みです。
9. ノーマル割込みステータスレジスタの DMA 割込みビット (SDHC_CORE_NORMAL_INT_STAT_R.DMA_INTERUPT) をクリアするために、'1'を書き込みます。
10. システムアドレスレジスタ (SDHC_CORE_SDMASA_R または DHC_CORE_ADMA_SA_LOW_R) を更新して、ステップ 7 に進みます。
11. ノーマル割込みステータスレジスタ (SDHC_CORE_NORMAL_INT_STAT_R) のデータ転送完了割込みビット (XFER_COMPLETE) と DMA 割込みビット (DMA_INTERUPT) をクリアするために、'1'を書き込みます。

2.3 アドバンスド DMA (ADMA) 使用時のデータ転送シーケンス

ADMA 使用時のデータ転送シーケンスを Figure 3 に示します。

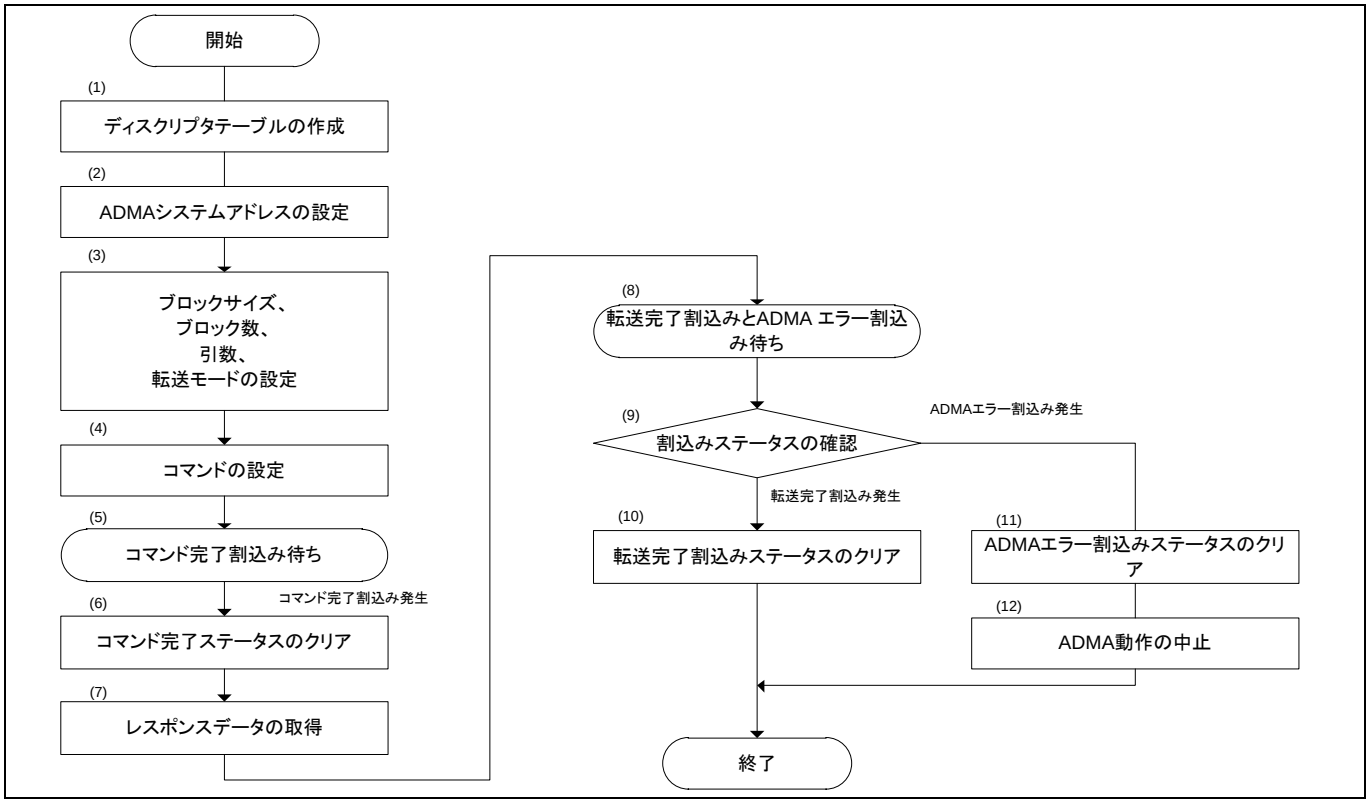

Figure 3 アドバンスド DMA (ADMA) 使用時のデータ転送シーケンス

上記のフローチャートで番号表示されている箇所を以下に説明します。

1. システムメモリ内に ADMA のディスクリプタテーブルを作成します。
2. ADMA システムアドレスレジスタ (SDHC_CORE_ADMA_SA_LOW_R) に ADMA のディスクリプタアドレスを設定します。
3. 以下の設定をします。
   a) 必要なブロックの長さをブロックサイズレジスタ (SDHC_CORE_BLOCKSIZE_R) に設定します。
   b) 必要なデータブロック数をブロックカウントレジスタに設定します。
   SDHC_CORE_HOST_CTRL2_R.HOST_VER4_ENABLE = ‘0’または 16 ビットブロックカウントレジスタ (SDHC_CORE_BLOCKCOUNT_R) に 0 以外が設定されている場合、16 ビットブロックカウントレジスタが選択されます。SDHC_CORE_HOST_CTRL2_R.HOST_VER4_ENABLE = ‘1’かつ 16 ビットブロックカウントレジスタに 0x0000 が設定されている場合、32 ビットブロックカウントレジスタ (SDHC_CORE_SDMASA_R) が選択されます。
   c) コマンドフォーマットのビット 8～39 で指定されているコマンド引数を引数レジスタ (SDHC_CORE_ARGUMENT_R) に設定します。
   d) 転送モードレジスタ (SDHC_CORE_XFER_MODE_R) を設定します。ホストドライバは、マルチ/シングルブロック選択, ブロックカウント有効, データ転送方向, Auto CMD12 有効, および DMA 有効を決定します。レスポンス確認が有効 (RESP_ERR_CHK_ENABLE = ‘1’) な場合、レスポンス割込みディセーブル (RES_INT_DISABLE) を‘1’に設定します。そして、レスポンスタイプ (RESP_TYPE) は、R1 (0: メモリ) または R5 (1: SDIO) を選択します。
4. 実行するコマンドをコマンドレジスタ (SDHC_CORE_CMD_R) に設定します。

5. レスポンス確認を有効にしている場合は、ステップ (8) に進みます。そうではない場合は、コマンド完了割込みを待ちます。
6. ノーマル割込みステータスレジスタのコマンド完了割込みビット (SDHC_CORE_NORMAL_INT_STAT_R.CMD_COMPLETE) をクリアするために、‘1’を書き込みます。
7. 発行されたコマンド対する必要なレスポンスを取得するためにレスポンスレジスタ (SDHC_CORE_RESPXX_R) を読み出します。
8. 転送完了割込みと ADMA エラー割込みを待ちます。
9. 転送完了ビット (SDHC_CORE_NORMAL_INT_STAT_R.XFER_COMPLETE) が‘1’の場合、ステップ 10 に進みます。そうではなく ADMA エラー割込みビット (SDHC_CORE_ERROR_INT_STAT_R.ADMA_ERR) が‘1’の場合、ステップ 11 に進みます。
10. ノーマル割込みステータスレジスタの転送完了割込みステータスビット (SDHC_CORE_NORMAL_INT_STAT_R.XFER_COMPLETE) をクリアするために、‘1’を書き込みます。
11. エラー割込みステータスレジスタの ADMA エラー割込みステータスビット (SDHC_CORE_ERROR_INT_STAT_R.ADMA_ERR) をクリアするために、‘1’を書き込みます。
12. ADMA 動作を中止します。SD カードの動作は、アボートコマンドを発行して停止する必要があります。ホストドライバは ADMA エラーステータスレジスタ (SDHC_CORE_ADMA_ERR_STAT_R) を確認して、発生した ADMA エラーを検証できます。

3 アボート転送シーケンス

アボート転送は、SD カードに対して CMD12 を発行し、SDIO に対して CMD52 を発行することによって実行されます。ホストドライバがアボート転送を実行する必要があるケースは 2 つあります。1 つ目のケースは、ホストドライバが無限ブロック転送を停止する場合です。2 つ目のケースは、マルチブロック転送中にホストドライバが転送を停止する場合です。

アボートコマンドを発行するためには、2 つの方法があります。

- **非同期アボートシーケンス:** Present ステートレジスタのコマンド禁止 (SDHC_CORE_PSTATE_REG.CMD_INHIBIT) が‘1’に設定されていない限り、ホストドライバはいつでもアボートコマンドを発行できます。
- **同期アボートシーケンス:** ブロックギャップ制御レジスタのブロックギャップ転送停止要求ビット (SDHC_CORE_BGAP_CTRL_R.STOP_BG_REQ) を‘1’に設定し、データ転送を停止した後、ホストドライバは、アボートコマンドを発行します。

3.1 アボートコマンドシーケンス

アボートコマンドシーケンスを Figure 4 に示します。

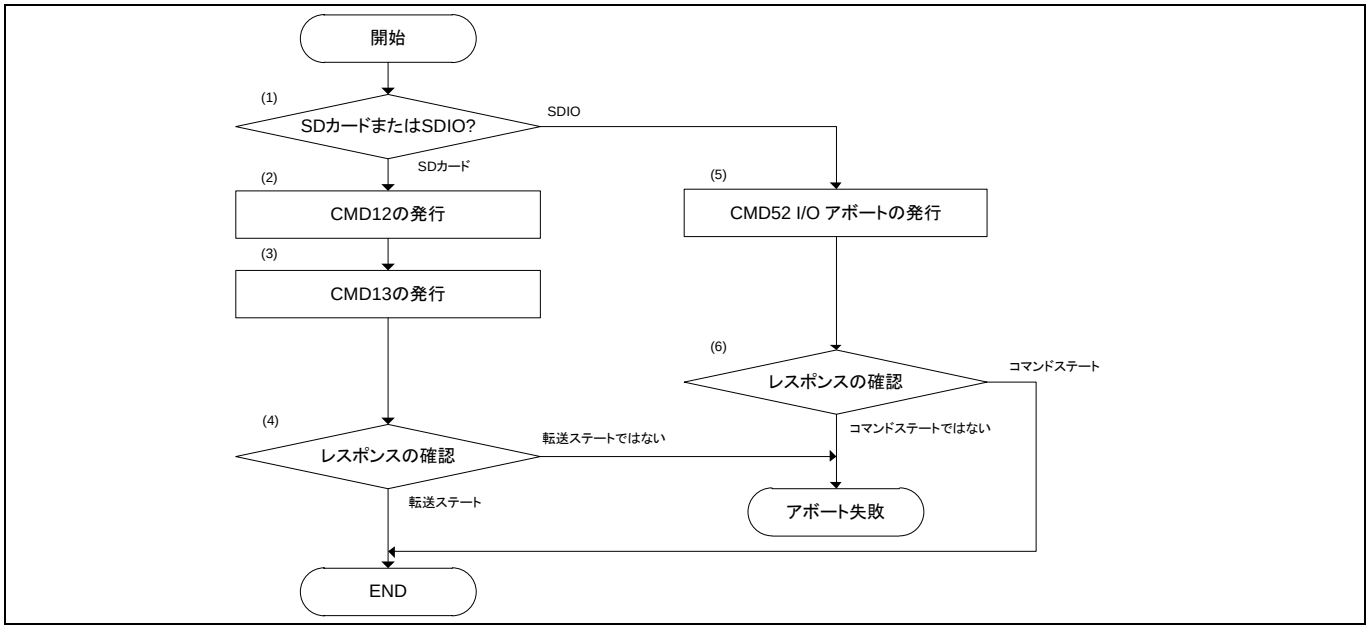

Figure 4 アボートコマンドシーケンス

上記のフローチャートで番号表示されている箇所を以下に説明します。

1. 接続されているデバイスが SD カードか SDIO かを確認します。ステップ 2 から 4 はカードアボート、ステップ 5 から 6 は I/O アボートのシーケンスです。
2. カードアボートのために CMD12 を発行します。カードが既に転送ステートの場合、CMD12 は受け入れられず、ホストドライバは、ステップ 3 でカードステートを確認する必要があります。
3. CMD12 の完了後、カードのステートを確認するために CMD13 を発行します。
4. レスポンスを確認し、カードが転送ステートの場合、アボートは成功です。そうではない場合、アボートは失敗です。
5. CMD52 I/O アボートを RAW (read after write) で発行します。
6. レスポンスを確認します。SDIO カードがコマンドステートの場合、アボートは成功です。そうではない場合、アボートは失敗です。

3.2 非同期アボートシーケンス

非同期アボートシーケンスを Figure 5 に示します。

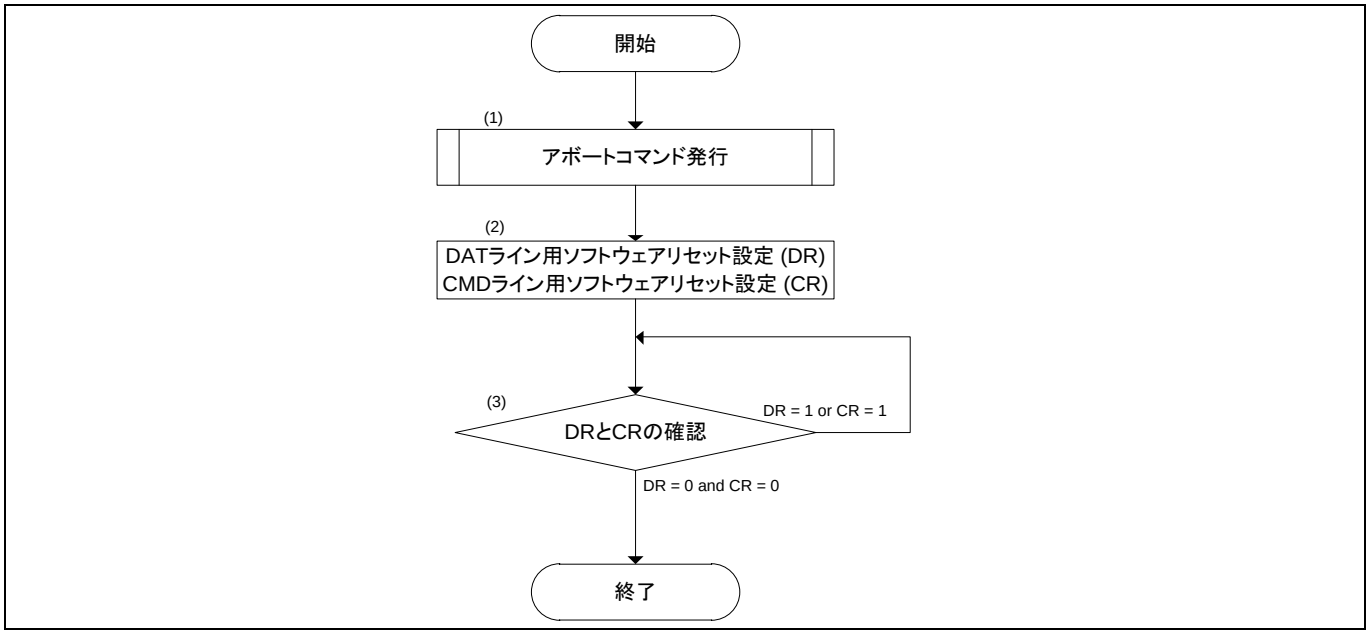

Figure 5 非同期アボートシーケンス

上記のフローチャートで番号表示されている箇所を以下に説明します。

1. アボートコマンドを発行します。
2. ホストコントローラバッファ内のデータを破棄するために、ソフトウェアリセットレジスタ (SDHC_CORE_SW_RST_R) の DAT ライン用ソフトウェアリセット (SW_RST_DAT) を‘1’に設定します。ステップ (1) のアボートコマンドが正常に完了した場合、ソフトウェアリセットレジスタ (SDHC_CORE_SW_RST_R) の CMD ライン用ソフトウェアリセット (SW_RST_CMD) によるコマンド回路リセットは不要です。
3. ステップ (2) で実行したソフトウェアリセットが完了するのを待つために、ステップ 2 で‘1’に設定したビットが‘0’にクリアされるまで待ちます。

3.3 同期アボートシーケンス

同期アボートシーケンスを Figure 6 に示します。

3 アボート転送シーケンス

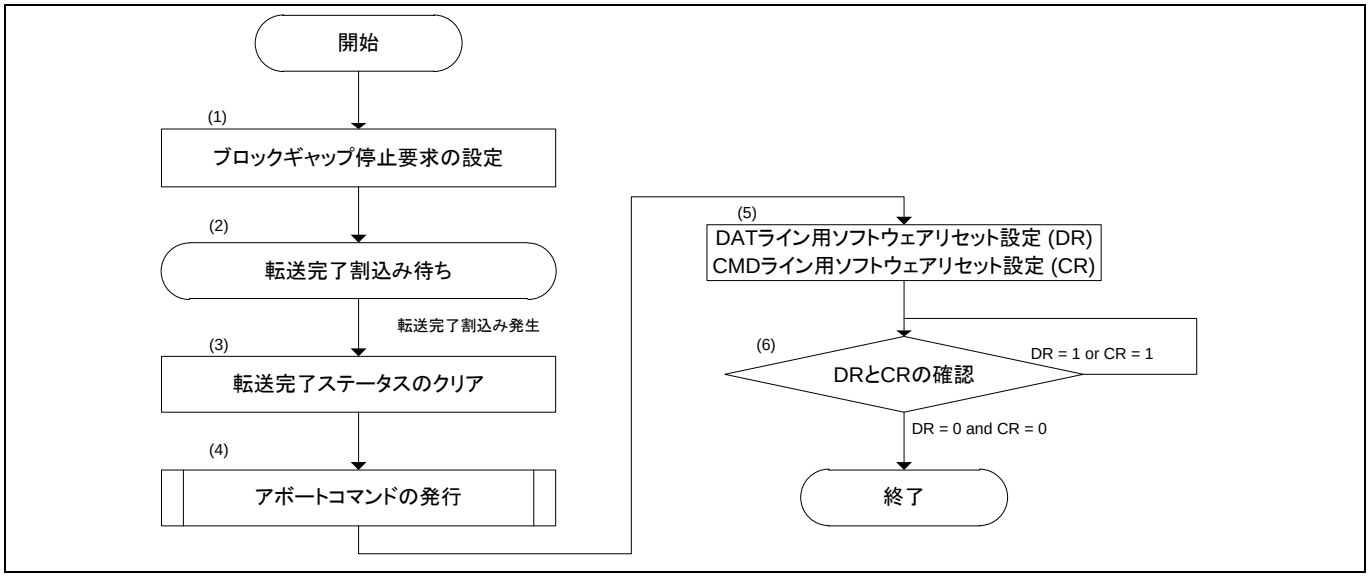

Figure 6　同期アボートシーケンス

上記のフローチャートで番号表示されている箇所を以下に説明します。

1. ブロックギャップ制御レジスタのブロックギャップ転送停止要求ビット (SDHC_CORE_BGAP_CTRL_R.STOP_BG_REQ) を‘1’に設定して、SD 転送を停止します。
2. 転送完了割込みを待ちます。
3. ノーマル割込みステータスレジスタの転送完了ビット (SDHC_CORE_NORMAL_INT_STAT_R.XFER_COMPLETE) をクリアするために、‘1’を書き込みます。
4. アボートコマンドを発行します。読出し動作を停止するために SD クロックが停止されている場合、ホストコントローラは、アボートコマンドを発行できるようにするために SD クロックを供給しますが、DMA を含むデータ回路は停止したままです。
5. ホストコントローラバッファ内のデータを破棄するために、ソフトウェアリセットレジスタ (SDHC_CORE_SW_RST_R) の DAT ライン用ソフトウェアリセット (SW_RST_DAT) を‘1’に設定します。ステップ 4 のアボートコマンドが正常に完了した場合、ソフトウェアリセットレジスタ (SDHC_CORE_SW_RST_R) の CMD ライン用ソフトウェアリセット (SW_RST_CMD) によるコマンド回路リセットは不要です。
6. ステップ 5 で実行したソフトウェアリセットが完了するのを待つために、ステップ 5 で‘1’に設定したビットが‘0’にクリアされるまで待ちます。

3.4　リセットコマンド

ホストとカード間の通信がアボート転送によって回復されない場合、ホストドライバは、リセットコマンド (CMD0) を使用する必要があります。リセットコマンドを発行する前に、DAT ライン用ソフトウェアリセットと CMD ライン用ソフトウェアリセットを実行して、ホストコントローラのコマンドおよびデータ回路をリセットします。

関連ドキュメント

以下は TRAVEO™ T2G ファミリ シリーズのデータシートとテクニカルリファレンスマニュアルです。これらのドキュメントについては、テクニカルサポートにお問い合わせください。

- デバイスデータシート
  - CYT3BB/4BB datasheet 32-bit Arm® Cortex®-M7 microcontroller TRAVEO™ T2G family
  - CYT4BF datasheet 32-bit Arm® Cortex®-M7 microcontroller TRAVEO™ T2G family
  - CYT6BJ datasheet 32-bit Arm® Cortex®-M7 microcontroller TRAVEO™ T2G family (Doc No. 002-33466)
- Body controller high ファミリ
  - TRAVEO™ T2G automotive body controller high family architecture reference manual
  - TRAVEO™ T2G automotive body controller high registers architecture reference manual for CYT3BB/4BB
  - TRAVEO™ T2G automotive body controller high registers architecture reference manual for CYT4BF
  - TRAVEO™ T2G automotive body controller high registers architecture reference manual for CYT6BJ (Doc No. 002-36068)

用語集

Table 1　用語集

用語	説明
ADMA	アドバンスド DMA
Auto CMD12	自動的に CMD12 を発行してマルチブロック転送動作をアボートするホストコントローラ機能
Block	バイト数, 基本データ転送単位
Block Gap	データブロック間の期間
CMD	コマンド
CMD12	マルチブロック/無限ブロック転送動作をアボートするためのコマンド
DAT	データバス
eMMC	Embedded Multi-Media Card
SD	Secure Digital
SDHC	Secure Digital High Capacity
SDIO	Secure Digital Input Output
SDMA	シングルオペレーション DMA

改訂履歴

版数	発行日	変更内容
**	2019-10-09	これは英語版 002-24414 Rev. **を翻訳した日本語版 Rev. **です。
*A	2020-04-28	これは英語版 002-24414 Rev. *A を翻訳した日本語版 Rev. *A です。
*B	2021-10-15	テンプレートの変更を実施。 これは英語版 002-24414 Rev. *B を翻訳した日本語版 Rev. *B です。
*C	2026-03-04	これは英語版 002-24414 Rev. *C を翻訳した日本語版 Rev. *C です。

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2026-03-04

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2026 Infineon Technologies AG. All Rights Reserved.

Do you have a question about this document?

Email: erratum@infineon.com

Document reference

002-28580 Rev. *C

重要: 注意事項

本文書の全ての日本語情報は、英文で提供されている情報の参考訳です。英文版と和文版に齟齬がある場合は常に英文版の内容が優先します。

製品 (サンプルも含みます。また、ハードウェア、ソフトウェアまたはその両方で構成される製品も含みます。以下同じです。) の販売、提供または引渡しは、インフィニオンテクノロジーズAGおよびその関連会社 (以下まとめて「インフィニオン」といいます)、お客様とインフィニオンによって締結された基本契約その他の書面により合意された条件、そのようなの条件がない場合はインフィニオンの適用ある販売条件に従います。お客様の一般的な取引条件またはインフィニオンの販売条件と乖離した条件は、インフィニオンが書面で明示的に同意した場合に限り、インフィニオンを拘束するものとします。
疑義を避けるため、インフィニオンは、第三者の権利を侵害していないことの保証、および特定の使用/目的に対する適合性または商品性の保証などの黙示的保証を負いません。
インフィニオンは、サンプル、アプリケーション、もしくはお客様による製品の具体的な使 用方法、または本書面に記載された例や標準値に関するいかなる情報についても責任いません。
本書面に記載されているデータは、技術的資格を有する熟練したお客様のご担当者をのみを対象としています。お客様は、意図された用途およびお客様の特定用途に対する製品の適合性を評価し、意図された用途および顧客の特定用途において本書面に含まれるすべての関連技術データを検証する責任を負います。お客様は、意図されたアプリケーションの機能性および安全性を適切に設計、プログラミング、テストし、その使用に関連する法令を遵守してください。
インフィニオンの明示的な承認がない限り、製品は、製品の故障またはその使用による結果が人身傷害につながると合理的に予想されるいかなる用途にも使用しないでください。ただし、上記は、インフィニオンが明示的に設計し、販売した使用分野でお客様が製品を使用することを妨げるものではありません。
インフィニオンは、適用される法律、例えばドイツ著作権法 (UrhG) 第44b条に従って、そのコンテンツを商業的なテキストおよびデータのマイニング (TDM) のために使用する権利を明示的に留保します。
製品にセキュリティ機能が含まれている場合:
いかなるコンピューティング・デバイスも絶対的に安全であることはあり得ないため、製品にセキュリティ対策が施されているにもかかわらず、インフィニオンは、製品に侵入、データの盗難もしくは損失、またはその他の侵害 (「セキュリティ侵害」といいます) がないことを保証せず、インフィニオンは、セキュリティ侵害に起因するいかなる責任も負わないものとします。
本書面がソフトウェアを含んでいるまたはソフトウェアに言及している場合：
ソフトウェアは、米国、ドイツ、および世界各国の知的財産権法および条約に基づきインフィニオンが所有しています。インフィニオンはすべての権利を留保し、無断複写、複製および転写を禁じます。お客様はソフトウェアに同梱されたソフトウェア使用許諾契約に従ってのみソフトウェアを使用できます。
ソフトウェア使用許諾契約が適用されない場合、インフィニオンはここに、ソフトウェアの知的所有権に基づき、(a) ソースコード形式で提供されたソフトウェアについて、インフィニオンのハードウェア製品と共に使用する目的のためにのみソフトウェアを修正および複製するための、そして(b)インフィニオンのハードウェア製品上での使用目的に限り、ソフトウェアをバイナリコード形式でエンドユーザーに外部配布するための、個人的、非排他的、譲渡不能なライセンス (サブライセンス不可) をお客様に付与します。それ以外のソフトウェアの使用、複製、変更、翻訳またはコンパイルは禁止されています。製品、技術、納品条件、および価格に関する詳細については、最寄りのインフィニオンオフィスに連絡するか、https://www.infineon.comにアクセスしてください。