



Serial ATA 規格のポイントと今後の動向

菅原 博英

Serial ATA 規格は、シリアル化した新しい ATA 規格である。従来の ATA (IDE) 規格とソフトウェア互換性をもっているものの、仕様書を見ると、IDE とはまったく異なる点も多い。

本章では、この Serial ATA 規格のポイントについて、IDE との相違点に注目しながら解説を進めていく。また、今後の動向についても触れる。(編集部)

はじめに

現在、HDD は PC のもっとも基本的かつ必須なデバイスの一つです。そのインターフェースである ATA (AT Attachment) 規格は、PC の I/O インターフェースの構成やロードマップに少なからず影響を与え、かつ与えられる要素です。

パラレルからシリアルへ変化し、高速化した ATA 規格である Serial ATA (以下、SATA) も、そのような背景の中から誕生しました。

踏み込んだ内容については、本章の後半で触れますが、Serial ATA は現在の PC で大成功を収めた IDE (Integrated Drive Electronics。従来のパラレルの ATA。最近では Serial ATA に対して、Parallel ATA や PATA と呼ぶ) をスムーズに先端的なインターフェースに生まれ変わらせ、移行させるかという命題を背景に誕生したといえます。

Serial ATA は、1999 年ごろ、Intel 社および APT technologies 社 (米国のカリフォルニア州サンタクルーズにあった Seagate 社のスピナウト、現在は Vitesse 社) を中心メンバとした SATA Working Group により仕様が検討されてきました。日本からは Steering (Board) メンバとしての参加はありませんでしたが、富士通などの HDD ベンダが当初からの Contribution メンバとして仕様策定に貢献してきました。

SATA は、IDE 特有の問題や、PC 用としてのコスト問題などを考慮した、いくつかの“コロンブスの卵”的なアイデアと割り切りで成功したインターフェースとして、成長途上にあります。

本章では、この Serial ATA (Serial ATA 1.0a 版) の仕様について解説していきます。仕様書は、下記の Web サイトからダウンロードできます。

<http://www.serialata.org/specifications.asp>

IDE については詳しく書かれた書籍などがすでにあるため、本章では読者の方々がある程度 IDE (PATA) に関する知識を持っていることを前提に、おもに IDE と SATA の違いに注目

して説明を進めていきます。

また、インターフェースの仕様は物理層 (信号層) から説明を開始するケースが多く見られますが、ここではコマンド層、すなわちホスト側のソフトウェア、あるいはデバイス側の上位コントロール層 (一般的にはファームウェア) からの視点で説明していきます。以下、とくに断りがない限り、ソフトウェアはホスト側のソフトウェアを、ファームウェアはデバイス側の上位コントロール層を指します。

① コマンド層

IDE のコマンド層から見た特徴は、すべての制御レジスタ (タスク・ファイル・レジスタ: Task File Register, TFR) が CPU のアドレス空間 (あるいは I/O ポート空間) にマッピングされていることです。CPU 上のソフトウェアは、このレジスタを直接リード/ライトしてデバイスを制御します。このタスク・ファイルがデバイス側のファームウェアからも同時に参照されます (図 1)。

この軽くシンプルなインターフェースが、CPU パワーが貧弱であった当初の PC で有効であり、そして成功した理由です。データの転送さえタスク・ファイルのデータ・レジスタをリード/ライトして行われていました (PIO モード)。

コマンド発行から完了までのタスク・ファイルへのアクセス方法は、当初はいろいろな (一般的でない) 使い方もあったかもしれませんが、しかし、一般的には、まずソフトウェアがタスク・ファイルに発行すべきコマンドのパラメーター式を書き込み、最後にコマンド・コードを書き込む、という方式です (図 2)。

ここで初めてファームウェアがタスク・ファイルに書き込まれた内容を参照してコマンドを実行します。コマンドがデータ転送を伴う場合、それもタスク・ファイルを経由してハンドシェイクを行い、最後にコマンド実行が完了したところでファームウェアが実行結果 (ステータス) をタスク・ファイルに反映し

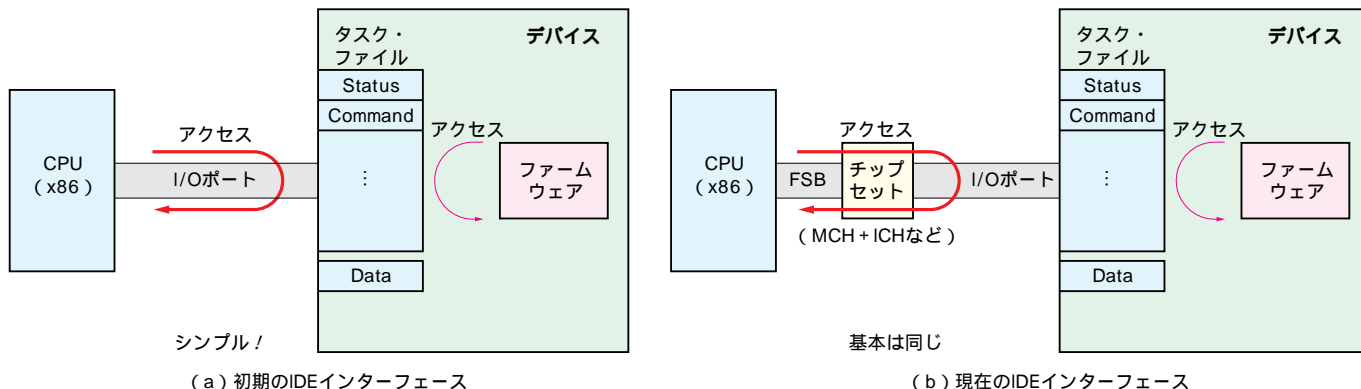


図1 IDE インターフェースの概念

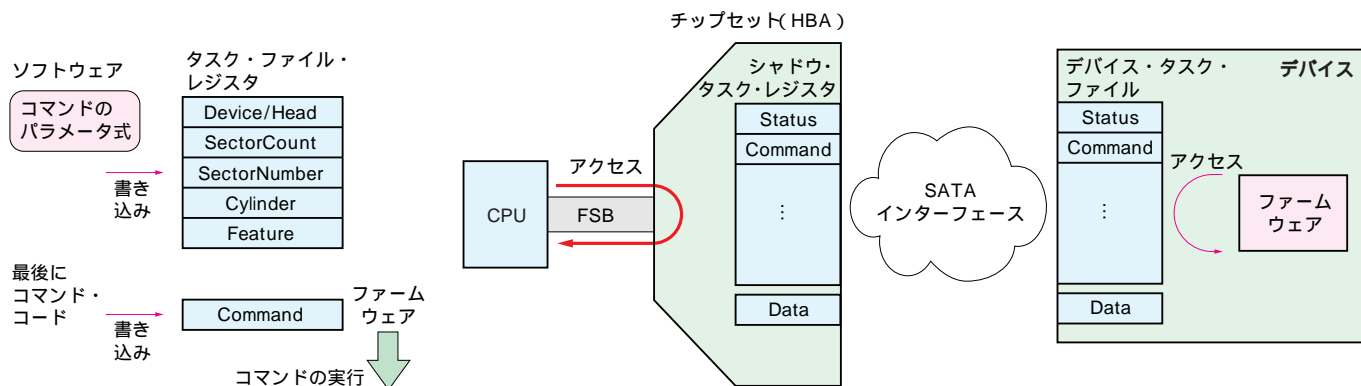


図2 IDE のコマンド実行

図3 Serial ATA における二つのタスク・ファイル

てコマンドを完了します。

コマンド発行前はステータスは 50h(Busy = OFF)であり、実行中は D0h(Busy = ON), 完了で元の 50h に戻ります。Busy 中は次のコマンドは発行できません^{注1}。

Serial ATA のコマンド・プロトコルは、この IDE の典型的なコマンド・シーケンスから発想されています。Serial ATA の場合、TFR は二つ存在することが概念上の大きな特徴となっています。すなわち、HBA(Host Bus Adaptor, ICH など)側に存在するシャドウ・タスク・ファイル・レジスタ(Shadow Task File Register : 以下、シャドウ TFR)とデバイス側の(本来の) TFR(区別のため、以下、デバイス TFR)です(図3)。

ソフトウェアからはシャドウ TFR のみが直接アクセス可能です。逆にファームウェアからは、デバイス TFR のみがアクセス可能です。この二つの TFR は、コマンド発行時と完了時に 2 種類の Register FIS(Frame Information Structure)によりお互いにコピーされます。また、PIO データ転送時など、必要に応じて両者の内容が一致するようにコピーが実行されます。

このコピー機能を、Fibre Channel やギガビット Ethernet などの通信制御と類似した階層に定義したものが Serial ATA 規

格の要点といえるでしょう。各階層は、以下のような一般的な通信制御のサブセットといえる、とてもシンプルな機能をもっています(図4)。

- 1) トランスポート層
コピーするためのフレームと順序(プロトコル)
 - 2) リンク層
フレームを転送するために、伝送路のアービトレーションとフロー制御, FIS, 制御コードをシリアル伝送しやすいビット列に変換(8b/10b 変換)
 - 3) 物理層
ビット列をシリアル・ビット化し、低振幅、ディファレンシャル信号として転送し受け取る
- 次に、各層について詳しく説明していきます。

② トランスポート層

フレーム(FIS : Frame Information Structure)
SATA では、従来の IDE をエミュレートするために以下の 8 個のフレームを定義しています。

注1 : 実行中でも PIO データ転送中は 58h(DREQ = ON)であり、そのほかの期間でもデバイスやコマンドによっては 58h である。ここでは簡単のため。