タイミングチャートを用いたデバイスドライバ開発自動化について

1J－2

島崎 要爾　落合 昭　大原 茂之

東海大学

1．はじめに

インタフェース（以下 I/F と略記）設計では，信号線の動作を記述するために，タイミングチャートが用いられることが多い．タイミングチャートは，時系列に沿った信号線の状態変化を図で表現したものであり，デバイスドライバ（以下ドライバと略記）を開発するには，タイミングチャートから処理手順を抽出する必要がある．しかし，タイミングチャートには統一された書式が存在せず，処理手順の抽出には経験を必要としてきた．

本稿では，タイミングチャートから処理手順を自動的に抽出し，ドライバを作成支援するシステムを提案する．

2．タイミングチャートの定義

タイミングチャートを定義するために必要な情報は，次の（1）～（4）の4項目である．

（1） I/F 仕様

I/F 仕様として必要な情報は，(I/F 名，通信方向，基本周期）の3項目である．I/F 名とは，ドライバの設計対象となる I/F の名称のことである．本稿では通信方向が，片方向通信の場合のみについて述べる．基本周期とは，タイミングチャートの動作の基本となるクロックの周期のことである．

（2） 信号線仕様

このシステムで扱う信号線には，データ線，アドレス線，制御線の3種類がある．

データ/アドレス線を定義するために必要な情報は，(信号線名，入出力方向，サイズ，論理，動作）の5項目である．信号線名とは，データ/アドレス線の名称のことである．入出力方向とは，送信側→受信側/受信側→送信側というデータ/アドレスの転送方向のことである．サイズとは，データ長/アドレス幅をビット数で示したものである．論理とは，正論理/負論理のことである．動作とは，通常動作/3ステート動作のことである．

制御線を定義するために必要な情報は，(信号線名，入出力方向，論理，種類），の4項目である．種類とは，ハンドシェーク用/メッセージ用という信号線の使用目的のことである．その他の項目はデータ/アドレス線仕様と同様である．

（3） 信号線状態

信号線状態とは，データ/アドレス線の場合「データ/アドレス出力」「非出力」の2状態，制御線の場合「Hi」「Lo」の2状態のことである．信号線の状態変化の最小単位は，基本周期である．

（4） 信号線状態変化のシーケンス

シーケンスとは，タイミングチャート上に記述された，処理の開始から終了までの信号線状態変化の順序と，データ送信側/受信側のどちらのドライバに必要な処理かを指定したものである．

3．ドライバ変換ルール

3.1 ドライバ生成手順概要

自動生成を行うドライバとして，データ送信側ドライバまたは受信側ドライバを選択する．作成するドライバにあわせて，信号線を操作する関数を作成する．加えて，作成するドライバに必要なシーケンスを，信号線操作関数の組み合わせで実現する．次節以降に，各関数の生成のルールを示す．

3.2 信号線操作関数作成ルール

データ出力/入力，状態読込み．変更といった信号線を操作する関数の作成を決定するために必要な情報は，(対象ドライバ，信号線入出力方向）の2項目である．対象ドライバとは，作成するドライバがデータ送信側か受信側のどちらであるかの情報である．信号線入出力方向とは，信号線仕様として定義された情報である．これら2つの情報より，それぞれの信号線を操作する関数の作成が決定される．

信号線操作関数は，「信号出力関数」「信号入力関数」の2種類である．それぞれの関数を作成するルールを表1に示す．

About Device Driver design method based on the Timing Chart.

Yoji SHIMAZAKI , Akira OCHIAI ,
Shigeyuki OHARA
Tokai University.

表1 信号線操作関数作成ルール

	対象ドライバ	信号線 入出力方向
信号出力関数	送信側	送信側→受信側
	受信側	受信側→送信側
信号入力関数	送信側	受信側→送信側
	受信側	送信側→受信側

信号線の種類が，データ/アドレス線の場合はデータ/アドレスが引数または戻り値となり，制御線の場合はHi/Loが引数または戻り値となる．

3.3 シーケンス関数作成ルール

入力されたシーケンスを処理手順に変換するために必要な情報は，(対象ドライバ，信号線種類，信号線入出力方向，信号線状態）の4項目である．作成ドライバ/信号線入出力方向は前節と同様である．ここでいう信号線種類は，データ/アドレス線，制御線の種類である．ここで，データ線とアドレス線は本質的に違いがないので，内部では同等に扱う．信号線状態は，シーケンスの過程にある，各信号線の状態のことである．

シーケンス関数は，シーケンスを開始から終了まで順番に調べていき，その過程における信号線と信号線の状態により，適宜信号線操作関数を追加していくことで作成する．

シーケンスの過程にデータ/アドレス線があり，データ/アドレス出力状態となっている場合の，データ/アドレス線操作関数追加ルールを表2に示す．

表2 データ/アドレス操作関数追加ルール

	対象ドライバ	信号線 入出力方向
データ/アドレス 出力関数	送信側	送信側→受信側
	受信側	受信側→送信側
データ/アドレス 入力関数	送信側	送信側→受信側
	受信側	受信側→送信側

データ/アドレス出力関数の追加では，出力するデータをシーケンス関数の引数として追加する．データ/アドレス入力関数の追加では，入力したデータをシーケンス関数の戻り値として追加する．

シーケンスの過程に制御線がある場合の処理の追加ルールを表3に示す．

制御線処理の追加では，信号線状態により，制御線状態変更処理では引数を，制御線状態待機処理では待機する状態を，適宜Hi/Loに切り替える．

以上のルールを用いて，チャート上に記述されたシーケンスをドライバに変換する．

表3 制御線処理追加ルール

	対象ドライバ	信号線 入出力方向
制御線状態 変更処理	送信側	送信側→受信側
	受信側	受信側→送信側
制御線状態 待機処理	送信側	送信側→受信側
	受信側	受信側→送信側

3.4 タイミングチャート入力例

図1に今回提案したシステムでのタイミングチャート入力例を示す．このタイミングチャートから自動的に作成したドライバの一部（シーケンス部分）を図1，図2に示す．

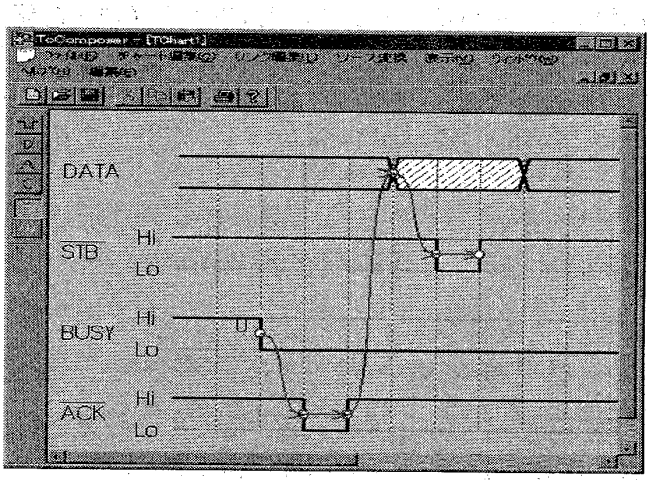

図1 タイミングチャート入力画面例

```
void Send_Data( int iData0 )
{
    while( Recv_BUSY() != LO);
    while( Recv_ACK() != LO);
    while( Recv_ACK() != HI);
    Send_DATA( iData0 );
    Send_STB( LO );
    Send_STB( HI );
}
```

図2 出力ドライバの一部

時間待ちの処理は機種依存部分が多いため，図2に示したドライバでは記述を省略した．

4．おわりに

本稿で提案したシステムで，タイミングチャートを用いてドライバ開発を自動化できることを示した．今後はI/Fを実現するハードウェアとドライバとの連携も含めた支援について検討していく予定である．