

7K-4

デバイスドライバ生成支援システム の設計について（3）

落合 昭 大原 茂之
東海大学

1. はじめに

デバイスドライバ（以下ドライバと略す）を設計するためには、その制御対象である周辺LSIやI/Fプロトコルといったものに関する知識が必要である。しかし、ハードウェアに関する知識や経験を持たないソフトウェア技術者にとって、それらの知識を獲得することは非常に困難である。そこで、設計者の知識や経験に依存することなく、ドライバを設計することができるデバイスドライバ設計支援システムを提案した¹⁾。

本報告では設計支援システムで用いられるモデルを設計する手順について述べる。

2. 設計支援システムの概要

2.1 ドライバのモデル化

ドライバはI/Fプロトコルに依存する部分と、LSIに依存する部分とに分けられる²⁾。前者をI/Fフレーム、後者をI/Fパーツと呼ぶ。図1参照。I/FフレームはI/Fプロトコルごとに存在し、ドライバが何をするかを表す。I/FパーツはLSIごとに存在し、I/Fフレームの各機能について具体的なLSIの操作方法を記述したものである。

I/Fパーツは、パーツ原型部、アドレス定義部、パラメータ定義部の3つの部分から構成される²⁾。パーツ原型部はLSIの使用条件が変わっても変化しない部分である。パーツ原型部で使用されるLSIのアドレスはアドレス定義部で定義され、LSIへのコマンドワードやマスク情報はパラメータ定義部で定義される。

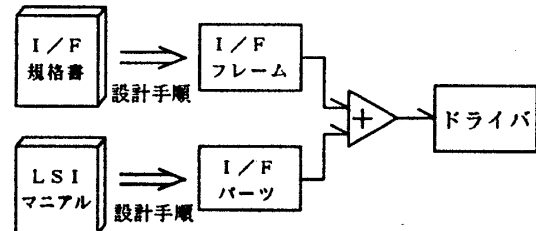


図1. ドライバ自動生成の原理

2.2 制御機能要素

I/Fフレームを構成する機能の中で、LSIを操作して機能を実現するものを制御機能要素（CFEと略す）とする。またCFEが実際にどうLSIを操作するかを記述したものをCFE実体と呼ぶ。CFEはCFE名によって識別される。

I/Fフレームを記述する時に、LSIの操作が必要な場合にはCFE名で記述する。I/FパーツはCFE実体の集合となる。I/FフレームとI/FパーツはCFEによって関係付けられる。

2.3 ドライバの合成

以下の手順によりドライバを生成することができる。

- ①対象とするI/F用のI/Fフレームと、対象とするLSI用のI/Fパーツを用意する。
- ②I/Fパーツ内のアドレス定義部およびパラメータ定義部に値をセットする。
- ③I/Fフレーム内のCFE名を、I/Fパーツの対応するCFE実体で置き換える。
- ④I/Fフレーム内のCFE名のすべてを置き換えれば、それがドライバとなる。

3. モデルの設計

3.1 I/Fパーツの設計手順

まずパーツ原型部の設計を行う。CFE名ごとにCFEの仕様にしたがって設計する。この時の設計ルールを「設計ルール1」に示す。

Design of a Support System for Generating
Device Drivers (3).

Akira OCHIAI, Shigeyuki OHARA.
Tokai University.

次に、パーツ原型部で用いられるアドレスを示す識別子と実際の値を対応させる定義式を、アドレス定義部に記述する。そして、パーツ原型部で用いられるコマンドを示す識別子およびパラメータを示す識別子と、実際の値を対応させる定義式を、パラメータ定義部に記述する。

[設計ルール1]

- A. LSIのレジスタを示すアドレスは、識別子を用いて記述し、この識別子とアドレス値の定義は別に行う。
- B. LSIへ送るコマンドワードは、識別子を用いて記述し、この識別子とコマンドワードの定義は別に行う。
- C. CFEに引数として渡されるデータがある場合には仮引数の名前と同じ名前の識別子を用いて引用する。
- D. 値を戻す場合には、CFE実体を式として記述する。
- E. LSIの使用方法により、変化する可能性のあるパラメータは直接記述せずに、識別子を用いて記述し、識別子と値とを別に定義する。

[ルール終わり]

(例) i8255の場合のI/Fパーツの一部を示す。@で始まるワードがCFE名である。その次のインデントされた部分がCFE実体でありLSIの具体的な操作である。

A. パーツ原型部

```
@LSI_INIT
    outp( CMD_PORT, LSI_INIT_CMD );

@DATA_OUT , date
    outp( DATA_PORT, date );

@BUSY_IN
    ( inp(BUSY_PORT) & BUSY_MASK )
```

B. アドレス定義部

```
#define CMD_PORT    0x46
#define DATA_PORT  0x40
```

C. パラメータ定義部

```
#define LSI_INIT_CMD  0x81
#define BUSY_MASK     0x01
```

3. 2 I/Fフレームの設計手順

I/Fのプロトコルにしたがって設計する。基本的には、初期化、データの出力、データの入力の3つのルーチンを設計する。初期化ルーチンは、LSIの初期化および各信号の初期設定を行う。データ出力ルーチンは、データの出力を行う。データ入力ルーチンは、データの入力を行う。この時の設計ルールを[設計ルール2]に示す。

[設計ルール2]

LSIの操作は直接行わずに、必ずCFEを用いて行う。 [ルール終わり]

(例) セントロニクスI/Fの場合のI/Fフレームを示す。@で始まるワードがCFEである。

```
prn_init() {
    @LSI_INIT
    @STROBE_OFF
}

prn_output( int out_date ) {
    while( 1 ) {
        if( @BUSY_IN == 0 ) break;
    }
    @DATA_OUT( out_date )
    @STROBE_ON
    wait();
    @STROBE_OFF
}
```

4. おわりに

本報告ではI/FパーツおよびI/Fフレームの設計手順を示した。今後はこの手順のシステム化を進めて行く予定である。

謝辞

本研究を進めるに当たり、日頃お世話になっている本学海洋学部航海工学科鈴木常夫教授に感謝の意を表します。

参考文献

- 1) 落合, 大原: デバイスドライバ生成支援システムの設計について(2), 情報処理学会第47回全国大会, 6K-1, 1993
- 2) 落合, 田宮, 大原: デバイスドライバのモデル化に関する一提案, 情報処理学会第49回全国大会, 6M-5, 1994