

2B-2

NLPにおける文字制御の一技法

高本康明 村松昭男  
(富士通株式会社)

1. はじめに

コンピュータの日本語処理が進むにつれてNLP(日本語ラインプリンタ)の機能も同時に進歩した。現在、文字処理分野では一般に、文字種は英数字/JIS第一、第二水準漢字/罫線、文字向きは横書き、書体は明朝体、字体は全角/半角、文字サイズは12ポ/9ポ/7ポでの印字を実現している。

しかし、最近の要求は、縦書き、ゴシック体、1/4角文字など様々な文字形式に及んでいる。一方これらの要求を実現するには、印刷速度の低下やメモリの増加等の問題が予想される。

本稿では、これらの問題点を考慮して、今回開発したNLPの機能と制御方法について述べる。

2. NLPのシステム概要と印字機能

1) システム概要

NLPの主な構成は制御ファーム部、文字発生機構、印刷機構、ページメモリ、文字パターン、コード変換テーブル、文字展開情報テーブルからなる。(図1)

ホストからの印刷データは文字コードと制御コードからなり、文字パターンはフロッピーディスク又はホストからローディングする。

制御ファームは、ホストからの印刷データを処理し、文字発生機構に起動を掛けて、ページメモリにドットパターンを展開する。一ページ完成したところで、印刷機構に起動を掛けて印刷する。

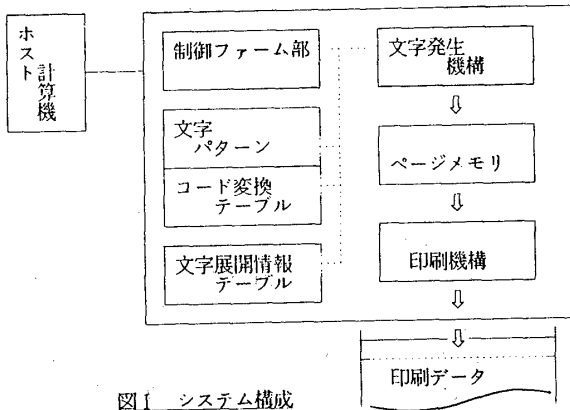


図1 システム構成

2) 印字機能

今回開発したNLPのサポート機能を、表1に示す。今回の機能拡張により印字形式が非常に増加し、印刷位置制御も含めると約1000種類にも及んだ。

表1 印字機能一覧

項目	文字向き	書体	字体	文字サイズ	文字ピッチ	行ピッチ	倍率
内容	縦書き	明朝体	全角	12ポ	5CPI	6LPI	一倍
	横書き	ゴシック体	半角	9ポ	6CPI	8LPI	二倍
			1/4角(上)	7ポ	8CPI	12LPI	長体
			1/4角(下)		10CPI		平体
					その他		

3. メモリ削減と処理速度向上の考え方

1) メモリ削減

縦と横、明朝とゴシックで共用可能なパターンは、どちらか一方の登録でも両方の印字が可能のようにし、文字パターン用メモリの縮小を計り、またコード変換テーブルの構成を変えることで、コード変換テーブル用メモリの縮小を計った。

2) 処理速度向上

文字パターンのページメモリへの展開は専用MPUにより高速文字展開する文字発生機構に役割分担し、制御ファームは文字パターンアドレスと展開原点及び展開情報を通知するのみとした。

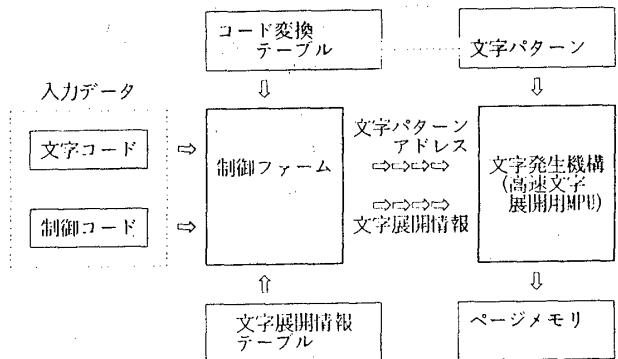


図2 制御ファームとハードの機能分担

4. メモリ削減と処理速度向上の実現方法

1) メモリ削減方法

a. パターンデータの縮小

縦書きパターン、ゴシックパターンが未登録の時は、それぞれ横書き、明朝パターンを使用することにし、同じパターンの二重登録を回避できるようにした。すなわち、拗音文字等の縦書き専用文字以外の縦書きパターンは横書きパターンのみの登録で済み、罫線パターンは明朝パターンでの登録で済むようにした。

b. 文字コード変換テーブルの縮小

図3のようなコード変換テーブルを作成した。

入力文字コードのJ E Fコードは、上位コード範囲がX"41" ~ X"FE" 下位コード範囲が X"A1" ~ X"FE"であることを考慮して上位と下位にテーブルを分け、存在する文字コードだけのテーブル構成にすることによって、従来のものと比べて3/4の大きさにすることができた。

また、テーブル内パターンアドレスは、パターンが128バイトの構成であることを考慮して実アドレスを7ビット右シフトした値でセットすることによって2バイト構成とした。

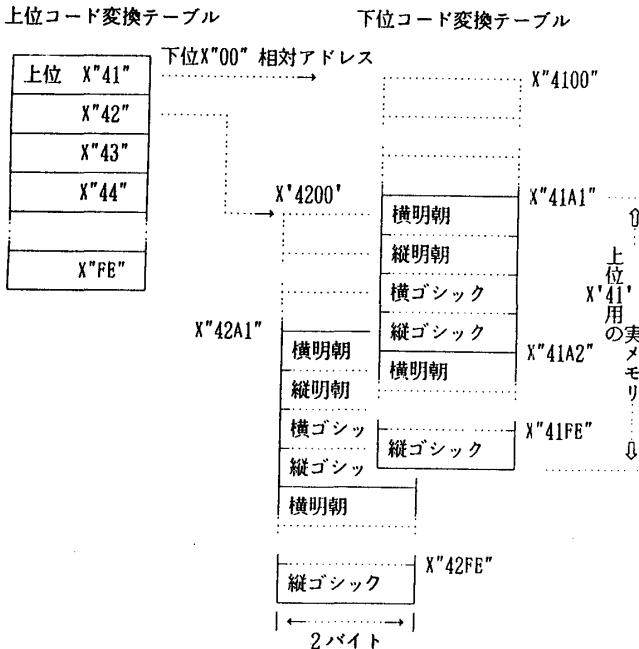


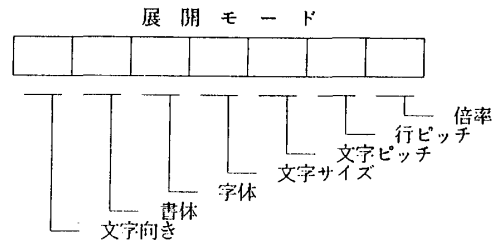
図3 コード変換テーブル構成

2) 速度向上方法

文字発生機構が文字パターンをページメモリーに展開するためには、文字パターンアドレス、展開位置、

展開幅、回転角度などを指定しなければならない。文字展開する場合、制御コードによって決まるこれらのパラメタをいかに早く文字発生機構に渡すことができるかによって、展開速度は大きく影響される。しいては、それによって印刷速度が決まる。そこで以下のような展開モードによる制御方法をとった。

展開モードの構成を図4に示すように、制御コード処理において指定された文字ピッチ、文字サイズ、倍率などのモード情報を直接に展開モードに表示しておく、そのデコード値をオフセットとして、展開情報テーブルより文字発生機構に渡すパラメタをすばやく得ることができるようにした。



文字展開情報 テーブル

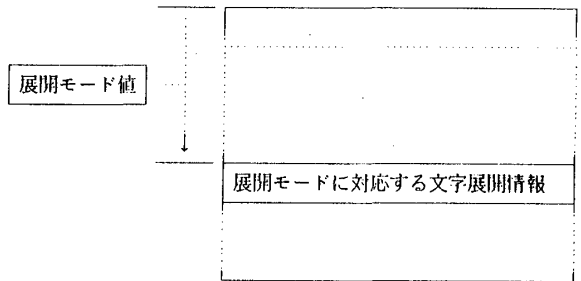


図4 展開モード構成と文字展開情報テーブルの構成

5. おわりに

今回の開発で考案した制御方法によって、データ処理のスピード化、メモリの削減はある程度実現できたと思う。

また制御コードが今後追加され、新たな文字種をサポートする場合でも、展開モード構成と展開情報テーブルを追加するだけで簡単に対応できることから拡張性も十分あるといえる。