

## 通信制御装置とマイクロプログラミング

山本 巖† 日比野 吉弘†

### 1. まえがき

通信制御装置 (Communication Control Unit; 以下 CCU と略す) は、データ通信システムにおいて、中央処理装置 (以下 CPU と略す) と、遠隔に散在する端末装置を通信回線で結んで、データの授受を行ない、その制御をするセンタ機器である。CCU の機能は、端末や回線に応じ、あるいは、ソフトウェア機能との分担によって変化するが、一般にコントロールは複雑である。ここでは、汎用の CCU に、マイクロプログラムを採用した例について述べ、その特徴、効果について解説する。

### 2. CCU の機能と特徴

本題に入る前に、CCU の機能と制御上の特徴について、簡単に触れる。

#### 2.1 機能

CCU の機能は、接続される CPU の性能 (命令の速度、種類、チャンネル特性など) に依存し、CPU のソフトウェアとの分担によって決まるが、次の (1)~(5) が代表的である。

- (1) 多数の回線 (又はモデム) と、CPU とのインタフェースを与える。
- (2) ビットサンプル、および文字の組立、分解を行なう。
- (3) 回線の接続と監視をする。
- (4) 誤り制御、伝送制御、コード交換のサポートをする。
- (5) CPU からの指令の解釈、実行、割込、状態の報告を行なう。

#### 2.2 制御上の特徴

CCU の制御面における特徴的な要素としては、次の (1)~(5) が挙げられる。

- (1) 回線側は多重化され、CPU 側は物理的に 1 チャンネルであるために、時分割多重の制御が必要であり、仕事の分割、時間の割当てを最適にする。すなわち、プログラムで実現すれば、能率の良いマルチタスク制御が必要である。

- (2) 数百にも及ぶ回線からのデータ、またはコントロールの要求を検出して、その回線アドレスを、迅速に識別しなくてはならない。すなわち、これを割込みで実現すれば、割込要求のある周辺機器アドレスを、高速に識別することが必要となる。
- (3) 時分割制御のため、回線毎の制御情報をメモリに貯えるが、その回線に属するメモリエリアには、速くアクセス出来ることが必要である。
- (4) 回線や端末を規定し、CCU の制御を回線毎に定めるパラメータ (例えば、回線速度、使用符号の単位数、パリティチェックの方式など) が非常に多く、その指定を判断したり、ビットからキャラクタへの組立て、あるいは内部の制御情報のセット、リセットと、判断のために、ビット操作とビット判断の機能が強化されていることが望ましい。これに対し、乗除算は不要で、加減算の重要性も低い。
- (5) 通信制御には、監視用 (無応答、回線断など) のタイマが不可欠であり、タイマ機能を必要とする。

### 3. CCU のマイクロプログラム

CCU の特徴を生かしてマイクロプログラミング技術がどのように応用されているかを述べる。

#### 3.1 ハードウェアとマイクロ命令

マイクロプログラム制御の CCU は、図 1 に示す機能ブロックから構成されるが、このような例では、ハードウェアによっては、高速回線におけるキャラクタ

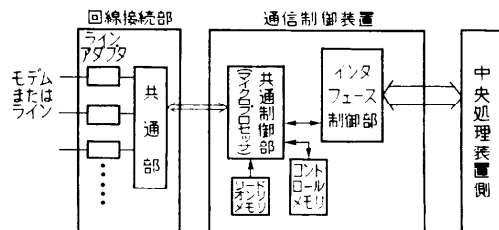


図 1 CCU の構成

Fig. 1 Block Diagram of Communication Control Unit.

† 日本電気株式会社周辺端末機器本部通信端末技術部

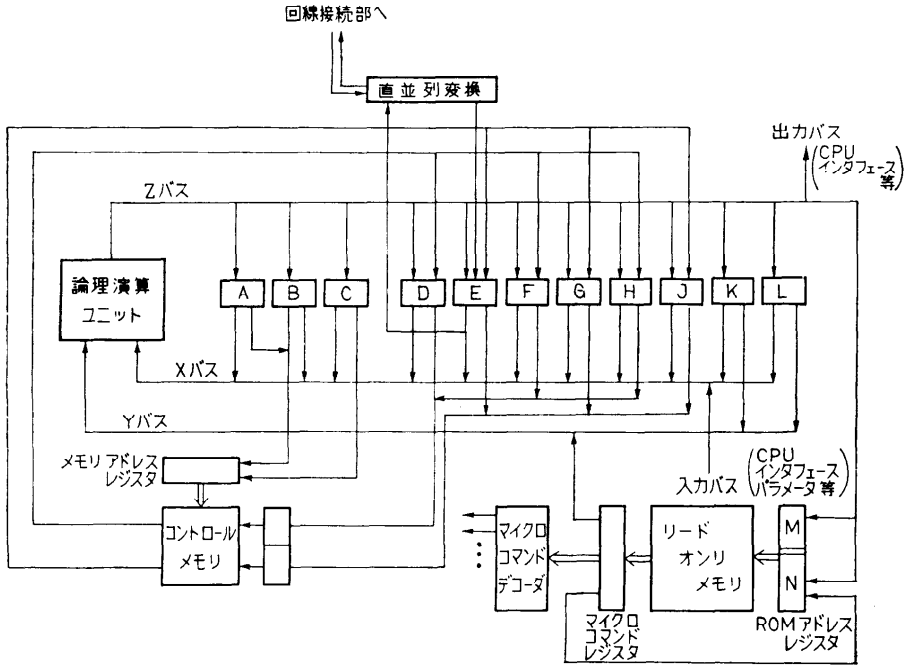


図 2 CCU のマイクロプロセッサのブロック図  
Fig. 2 Block Diagram of CCU Micro Processor.

の分解, 組立機能, 低速回線におけるビットサンプリング (以上回線接続部), CPU とのデータ転送, 指令や状態情報の転送と CPU への割込み信号発生, および CCU 全体を制御する指令の実行 (以上インタフェース制御部) までを分担し, マイクロプログラムによっては, 低速回線のキャラクタ分解, 組立て, 回線毎に実行すべき指令の実行とデータ転送の要求発生などを分担している. マイクロプロセッサの構成は, 図2に示すようなバス構造で, マイクロ命令は, 1語 42ビットから成り, サイクルタイム  $1\mu\text{s}$  のリードオンリメモリに収容されている. 制御情報やキャラクタバッファなどを貯えるラインコントロールワードのメモリとしては, サイクルタイム  $2\mu\text{s}$  のコアメモリを使っている. したがって, コアメモリへの1回のアクセス中に, 2つのマイクロ命令を実行できる. マイクロ命令の形式は図3に示すが, このうち SA, SB, C, OP, DT がバスオペレーション (バスと論理演算ユニットを使用する基本演算) を実行する部分である. SA, SB で指定されるレジスタの内容が, 図2の Xバス, Yバスを通して論理演算ユニットに入り, OP で指定される演算の後, DT が指定するレジスタへ, Zバスを通してセットされる. 演算の種類には, 論理積, 論理

和, 2進加算, ビットシフト, 排他的論理和がある. また C 指定をすることにより SB, CS フィールドが定数となり, SA で指定するレジスタと演算することができる. BR 1, BR 2 はテスト・ブランチコントロールで, 各種フリップフロップや外部制御信号の状態を検査し, NEXT と共に N レジスタの内容となる. M, N, 2つのレジスタによって, 次に実行すべきマイク

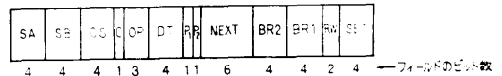


図 3 マイクロ命令形式  
Fig. 3 Format of Micro Command.

ロ命令のあるアドレスを選択する. SET フィールドと C 指定のない CS フィールドは, 各種フリップフロップや, 外部制御信号のセット, リセットを行なう. すなわち, BR はハードウェアからマイクロプログラムへの, また SET はマイクロプログラムからハードウェアへのインタフェースとなる. RW フィールドは, コントロールメモリの読出し, 書込みを制御し, アドレス指定は, A, B レジスタと SET, または C レジスタによって行なう. バスレジスタは, 各々 8ビットで, D~J はコントロールメモリの読出し書込みア

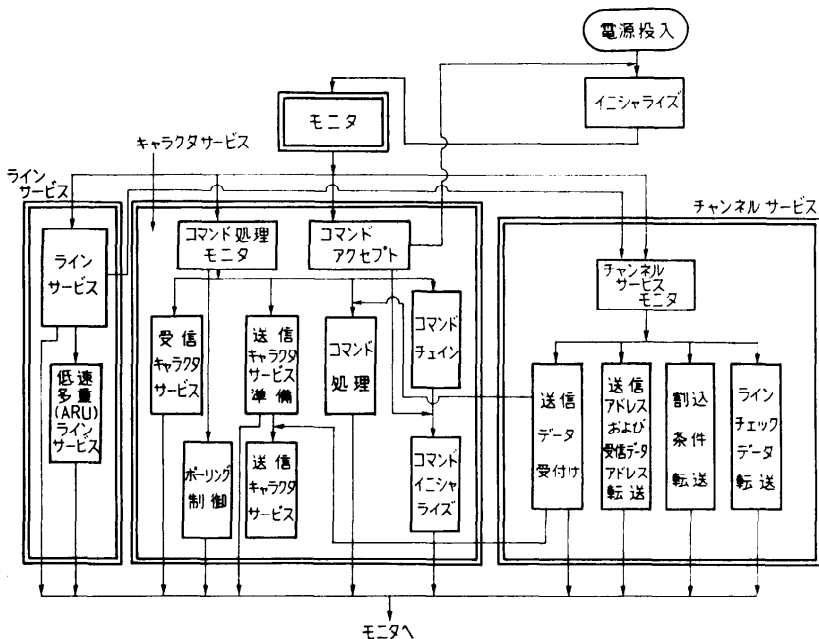


図4 マイクロプログラムの構成  
Fig. 4 Block Diagram of CCU Microprogram.

ータを入れるためと演算用，特にEレジスタは送受信データの直並列変換用で，K, L レジスタは作業用である。本マイクロプロセッサはメモリパリティエラー時のみ0番地へ割込むが，他の割込機能は持たない。

3.2 マイクロプログラム

CCU のマイクロプログラムの構成は，図4に示すが，大別して3つのサービス，すなわち3つの独立したプロセスから成る。ラインサービスは，回線対応部とのインタフェースを与え，キャラクタサービスは，キャラクタに対する処理と指令の実行，チャンネルサービスでは，CPU（実際にはCCUのインタフェース制御部）とのインタフェースを与える。この3つのサービス（の各プロセス）は，夫々独立した回線走査を行っており，これらを同期化するための制御情報は32バイトから成るラインコントロールワードの中に在る。ラインコントロールワードは，図5に示す要素で構成され，回線毎（全二重動作をするときは，2つのワードを必要とするが）に対応する。図4における各サービスをつくるルーチンのうち，主要ブロックの機能は，次の通りで，このために，全体でリードオンリメモリを約2,000ワード（但し，付加機構まで含む）使用している。

(1) モニタ；マイクロプログラムでコントロール

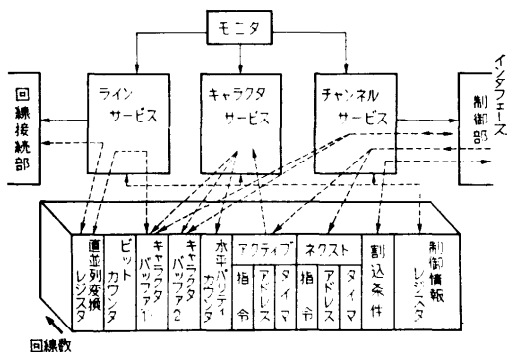


図5 ラインコントロールワードとマイクロプログラム制御  
Fig. 5 Microprogram Control Flow and Line Control Word.

できるスイッチフロップ，外部信号の状態により，次に処理すべき実行ルーチンを選択する。

- (2) ラインサービス；単位ビット時間毎に必ず入って，各回線を走査しつつ，要求があれば，文字の組立て，分解，プログラムタイマの更新，診断用送受信指令の実行をする。
- (3) コマンド処理モニター；回線を更新し，更新された回線の指令の種類にしたがって，次に行な

うべきルーチンを決め、コントロールを渡す。

- (4) 送信キャラクタサービス準備; 送信データの要求, 水平パリティ符号の送出, 応答割込符号の受け付けなどを行なう。
- (5) 送信キャラクタサービス; 送信データに対し, 水平パリティ計数, 制御符号の検出などの処理を行なう。
- (6) 受信キャラクタサービス; 受信データに対し, 水平パリティの計数, および照合, 垂直パリティのチェック, 制御符号の検出, 時間監視などを行なう。
- (7) 送信データ受け付け; 送信データをインタフェース制御部から受取る。
- (8) 送信アドレス転送および受信アドレス, データ転送; 送信データを要求するためにアドレスをインタフェース制御部に移送, または, 受信データとアドレスを移送する。
- (9) チャンネルサービスモニタ; 回線を更新し, 更新された回線に対して, 次に実行すべきルーチンを選ぶ。

なお, マイクロプログラミングのサポートとして, NEAC 2200 計算機によるマイクロプログラムアセンブラ MICAP (MICROprogram Assemble Program) と, インタプリタ形式の論理シミュレータ CPAS (Control Program Automation System) を用意した。アセンブラは, シンボリック言語からマシン言語に変換し, リードオンリメモリの配線用紙テープを出力する。また, CAPS は, 机上デバッグより正確に, 前もって論理シミュレーションすることを可能にした。

### 3.3 能力向上の工夫

CCU の回線収容率を上げるためには, 各サービスを出来る限り短時間に終えることが必要である。本 CCU では, 次の点を考慮し, 能力の向上に役立たせている。マイクロプログラミングの特徴は, ハードウェア, マイクロプログラム両面から, その装置に最も適した工夫を加えられる点にもあると思われる。

- (1) 2バイトの同時読出しと, 6つの読出し, 書込みレジスタ; ラインサービスでは, 1,200 ビット/秒の1ビット時間の内に, 多くの回線を走査し, ラインコントロールワードの情報を読出し, 判断し, 書込む必要がある。1バイト毎の読出しでは, メモリリミットになるので, 2バイト同時に読出すと共に, 6バイトの読出し書込みレジスタをもって, 退避する必要をなく

した(図2)。

- (2) 1メモリサイクル内でのビット更新機能; 先に述べたように, CCU の制御にビット操作性は, 特に重要である。したがって, 1メモリサイクル内で, 1ビットの変更だけを可能にし, この機能がなければ, 4マイクロステップ(書込完了までは5ステップ)要する動作を, 1マイクロステップ(書込完了までは2ステップ)で行なえる。
- (3) パラメータ(回線タイプの指定)検出の高速化と制御符号検出機能のハードウェア化; パラメータはメモリ内でなく, ジャンプ指定とし, この情報は BR, または SA 指定で Xバスに入れ, 直ちに読取ることができる。また, 制御符号検出機能は Fレジスタに入れた符号を BR でテストできるようにしている。
- (4) いくつかのパラメータ, あるいはレジスタの内容による多分岐の方法; DT で Nレジスタを指定できるので, コマンドの解釈や, 状態カウンタによるプログラムパスの選択は簡単である。このとき, NEXT フィールドをベースに分岐出来, 更に BR 機能も使える。
- (5) 回線とそれに属するメモリエリアへのアクセス; 回線アドレス指定と, それに属するラインコントロールワードのアドレス指定は, 同一レジスタで行ない, またワード内の 32 バイトにはマイクロ命令で直接指定ができる。

### 3.4 マイクロ命令の使用状況

標準機能を実現しているのは, 1481 マイクロステップであるが, 各マイクロコマンド別の使用状況は表1の通りである。この例では, マイクロ命令は, いくつかのマイクロファンクションを同時に指定できるので, 同じファンクションが異なる目的で使われる場合が多い。例えば, 論理和は, あるコントロールフラグをたてるのに使われたり, BR 機能と共に, そのフラグの有無を調べるのにも使われる。したがって, CCU の特徴を示すためには, 制御機能(目的)別, すなわち単位機能別の使用率で示するのが望ましいと思われる。等価単位機能別の使用率を, 表2に示した。これらの表によれば, 定数との演算が, 案外多いが, これは, レジスタやメモリへの初期値の設定に用いたり, 外部信号のセンスや, セットにも使われ, その他, 回線アドレスの走査, 転送先メモリのアドレス更新など, 多用途に使用されているためである。先に述べ

表1 マイクロコマンド別使用ワード数

分類	マイクロ・コマンド	使用ワード数
バス オペレーション (基本演算)	LOGICAL OR	651
	LOGICAL AND	195
	EXCLUSIVE OR	32
	BINARY ADD	85
	BIT SHIFT	6
セット コントロール		293
テスト ブランチ	BR1 フィールドのもの	238
コントロール	BR2 フィールドのもの	185
メモリ参照	1メモリサイクル内での1ビット更新	155
	読 出	165
	2バイト書込	88
	1バイト書込	122
定 数		891

た、1メモリサイクル内での1ビットの更新機能も、よく使われており、能力向上のみならず、リードオンリメモリのワード数の減少に役立っている。

#### 4. むすび

以上述べたように、我々は、CCU というオンライン制御装置にマイクロプログラミングを採用して、効果をあげてきた。本 CCU では、CCU の能力に比しリードオンリメモリのサイクルタイムもそれ程速くないこともあって、1ワードにいくつかのマイクロファンクションをもつ命令形式を採用したが、命令形式の如何を問わず、マイクロプログラミングの、一般的にいわれている利点は、CCU にもそのままあてはまる。しかし、計算機によるインタプリティブ形式のシミュレーションの試みは、CCU では、時間的に変化する入出力信号と、パラメータの数が多くて、準備にかかる時間と、シミュレーション時間が長くて、実用的でなかった。これは CCU が計算機のインストラクションのように、論理が閉じた系でないことにもよるのか

表2 単位機能別マイクロ命令使用頻度

分類	単 位 機 能	オペランド	使用率 (%)
オペレーション	レジスタ移送	定 数-レジスタ	16.1
	レジスタ移送	レジスタ-レジスタ	9.0
	NO OPERATION	—	6.5
	LOGICAL OR	定 数-レジスタ	3.6
	EXCLUSIVE OR	レジスタ-レジスタ	0.4
	BINARY ADD	定 数-レジスタ	4.1
	LOGICAL AND	定 数-レジスタ	3.6
	BIT SHIFT	レジスタ-レジスタ	0.6
シーケ ンスコ ントロ ール	比較ハブランチ	定 数-レジスタ	1.5
	Nレジスタ使用による多分岐	Nレジスタ	1.9
	パラメータ・センスハブランチ	定 数-入力バス	3.4
	レジスタビットセンスハブランチ	定 数-レジスタ	4.8
	ブランチ	(BR)	7.8
メモリ 参照	書 込	—	10.7
	読 出	—	8.5
	1メモリサイクル内の1ビット更新	—	7.6
セット コントロール		—	9.8

もしれない。今後は、速くて、再書込可能なメモリの出現で、シミュレーションは不要となり、素子の高速化、小型化と共に、益々マイクロプログラミングが使い易くなると思われ、データ通信機器や、制御機器の分野でも採用されていくであろう。

最後に、日頃御指導いただいている日本電気の関係各位に深謝する。

#### 参考文献

- 1) 山本 巖：データ通信，産報。
- 2) 山本，他：N 292 多重通信制御装置，NEC 技報，No. 93 (1969)。

(昭和48年3月23日受付)