

チャンネル結合によるネットワークシステムの構成

横山雅俊 小川 均 田中幸吉
(大阪大学・基礎工学部)

1. まえがき

近年、計算機メーカーより分散処理を目的とした価格/性能比の良いシステムが多く発表され、分散ネットワークシステムの実現も容易になった。阪大田中研究室においても、マイクロコンピュータやさまざまな周辺機器を含むシステム資源を有効利用するため、機能分散形ネットワークシステムを構成している。このようなネットワークシステムを構成する場合に最も重要な点は、計算機間結合(通信)方式である。本システムではミニコンピュータシステムに既存の標準的チャンネル(入出力アダプタも含む)を活用したチャンネル結合方式を用いている。ここでは、本システムの具体的な構成について紹介し、計算機結合における必要条件を挙げ、実際の運用に基づいてチャンネル結合方式の評価を与える。また、ターミナルのリモートコントロールを行うときのチャンネル結合の機能及び、柔軟なプログラム構造の実現を容易にするダイナミックリモートローディング方式についても検討する。

2. 計算機結合における必要条件

計算機結合においては次のような必要条件が望まれる。

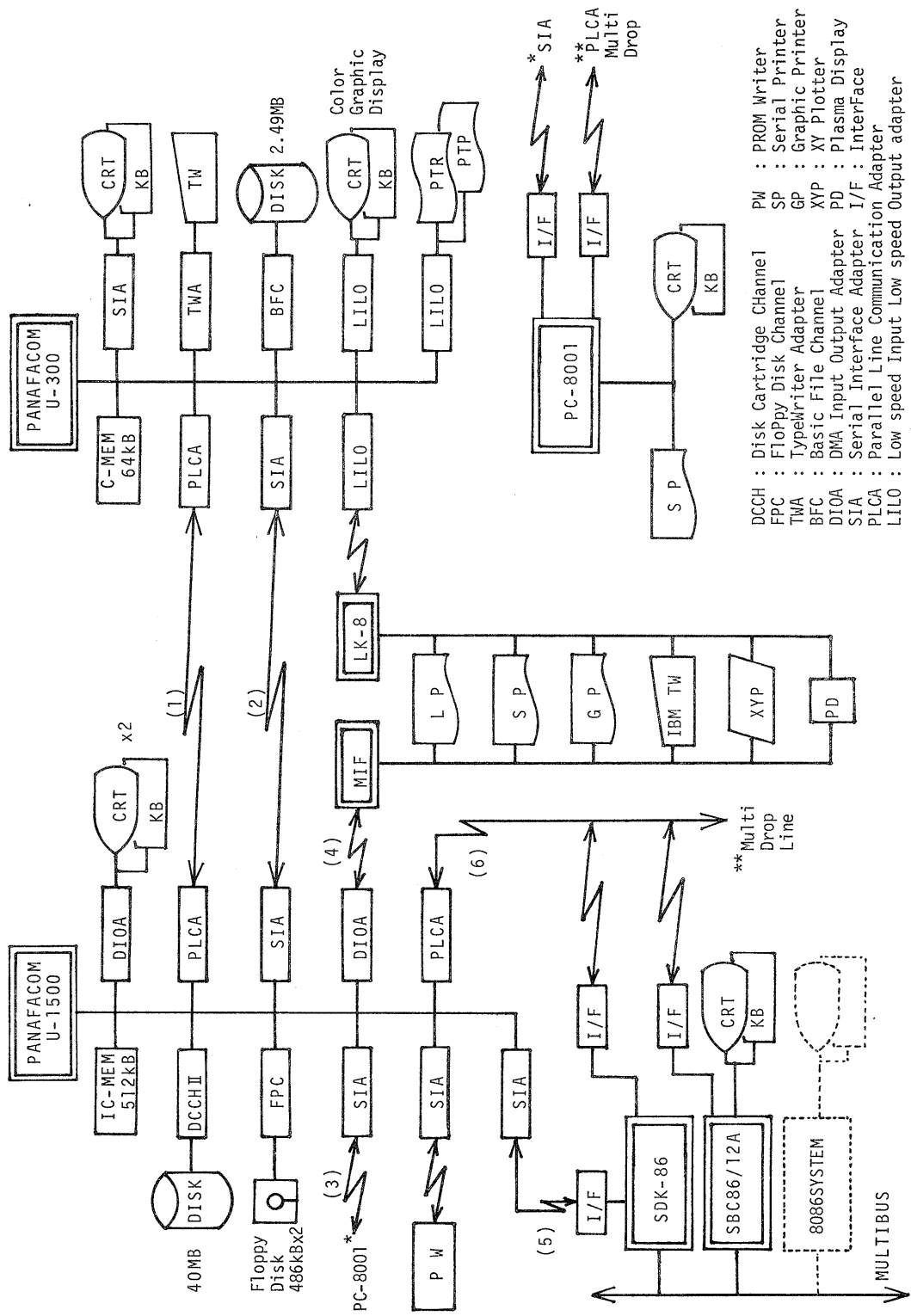
- (1) コストが小さい—安価なマイクロ(パーソナル)コンピュータをターミナルとして活用すること考えると、これに比較して結合コストを小さくしたい。
- (2) 高速転送が可能—大きなプログラムやファイルを転送するときには、当然高速転送が必要となる。ただし、ソフトウェア的なオーバーヘッドタイムを考えた場合、低速転送チャンネルの稼働率を上げる方が良いときもある。
- (3) 信頼性が高い—転送エラー発生時の検出、時間監視ができ、エラー発生時の制御手順が準備されている。
- (4) 拡張性がある—結合する相手が変更あるいは増設された場合の対応が容易にできる。信号レベル・タイミング等に汎用性があり、分岐接続時の信号衝突対策・ターミナル識別機構等がある。
- (5) 制御が容易である—汎用(ミニ)コンピュータが使用されている場合、そのOSや入出力制御プログラムを利用できるような構成にすれば、ソフトウェア(結合制御プログラム)の作成が容易になる。

3. 実験システム概要

図1に本システムのハードウェア構成を示す。本システムではPFU-1500をホストプロセッサとし、6台のターミナルプロセッサを使用している。ここでは各プロセッサの概略仕様及び本システムにおける機能について述べる。

3.1 PFU-1500

概略仕様	CPU	マイクロプログラム方式
	演算速度	レジスタ加算 0.45 μ s
	メモリ	IC-RAM 512 kB



- DCCH : Disk Cartridge Channel
- FPC : Floppy Disk Channel
- TWA : Typewriter Adapter
- BFC : Basic File Channel
- DIOA : DMA Input Output Adapter
- SIA : Serial Interface Adapter
- PLCA : Parallel Line Communication Adapter
- LILLO : Low speed Input Low speed Output adapter
- PW : PROM Writer
- SP : Serial Printer
- GP : Graphic Printer
- XYP : XY Plotter
- PD : Plasma Display
- I/F : Interface

図1 本システムのハードウェア構成

○本システムにおける機能

- ・ホストプロセッサとしてシステム全体を管理する。
- ・ファイルシステム(ターミナルの補助記憶装置の代行)。
- ・各種言語プロセッサ—FORTRAN・COBOL・LISP・アセンブラ(セルフ, M6800, I8086)・エディタ等

3.2 PFU-300

○概略仕様

CPU	マイクロプログラム方式
演算速度	レジスタ加算 0.8 μ s
メモリ	コア-RAM 64kB

- 本システムにおける機能—サブファイルシステム, 画像処理等を受け持つ。

3.3 LK-8

○概略仕様

CPU	MB8861 (M6800)
演算速度	レジスタ加算 2 μ s (8bit), 4 μ s (16bit)
メモリ	ROM 3kB, RAM 1kB

- 本システムにおける機能—非標準入出力装置(OSのサポートが無いもの)に対する制御を行う。複数の入出力装置が接続されており, それらのコード変換・タイミング制御等を選択的に実行する。

3.4 MIF

○概略仕様

CPU	MB8861
メモリ	ROM 8kB, RAM 8kB

- 本システムにおける機能—LK-8と同じ目的を持つ。どちらもセレクト4チャンネルとして動作しているため, 同時には各1つの入出力装置しか使用できない。現在, マルチプレクサチャンネルとして動作するための制御プログラムを検討中である。

3.5 PC-8001

○概略仕様

CPU	μ PD780 (Z80)
演算速度	レジスタ加算 1 μ s (8bit), 2.75 μ s (16bit)
メモリ	ROM 24kB, RAM 32kB
CRT	80 \times 25文字, 160 \times 100点, 8色カラー
シリアルI/F	TTLレベル, 300~4800 Baud

- 本システムにおける機能—PC-8001は本来BASIC言語を中心とした価格/性能比の良いパーソナルコンピュータであるが, 直列通信を行う基本的能力を持っており, インテリジェントターミナルとして活用することもできる。現在, 簡易手順を用いて4800 Baudの直列通信を行っているが, プログラマブルパリティフェーザルインターフェースLSI 8255を1個接続することにより, PLC A(後述)との分岐接続も可能となる。

3.6 SDK-86

○概略仕様

CPU	8086
演算速度	レジスタ加算 0.6 μ s, レジスタ転送 0.4 μ s
メモリ	ROM 8kB, RAM 4kB
シリアルI/F	RS232C 110~4800 Baud
パラレルI/F	48ライン(8255 \times 2)

- 本システムにおける機能—SDK-86は8086システムの評価試作用

に作られており、拡張が容易である。そのため、主に8086プログラムの開発に用いられ、突線動時には8086マルチマイクロプロセッサシステム全体を管理する。ホストプロセッサとは直列転送・並列転送両ルートを持ち、それぞれをどのように使い分けるかが重要な問題である。

3.7 SBC 86/12A

○概略仕様

CPU	8086
メモリ	ROM 8kB, RAM 32kB (Dual Port)
シリアル I/F	RS 232C 75~9600 Baud
パラレル I/F	24ライン(8255)
その他	割込制御(8259), タイマ 3組(8253) MULTIBUS インターフェース

○本システムにおける機能——これは高度な機能を持った1ボードマイクロコンピュータであり、マルチプロセッサ構成も容易にできるように、アドレスプリフィクス機構を備えている。本システムでは、今後このアドレスプリフィクス機構を持つ8086サブシステムを追加し、マルチプロセッシング機能を強化する予定である。

4. チャネル結合による通信機構

本システムで用いた結合用4チャンネル(DIOA, SIA, PLCA)は全てパナファコム製ミニコンピュータシリーズに用意されている標準的モジュールである。ここではPLCAを中心にそれぞれを説明する。

4.1 DIOA (低速入出力機器接続用インターフェース装置)

DIOAは本来コンソールカードリーダー・プリンタ等の標準入出力機器を接続するものであり、通常のシステムでは必ず持っている。このようなチャンネルを用いれば、ホスト側に特別なハードウェア/ソフトウェアを必要とせず、小さいコストで簡単に計算機結合ができる。さらに、コンソールインターフェースを利用できれば、全くホスト側に手を加えずにリモートジョブエントリが実現可能である。

4.2 SIA (モデム型シリアルインターフェース装置)

SIAはRS 232Cモデム型インターフェース(110~9600 Baud)を持っているが、DMA転送を行っている点で通常の通信回線制御装置と異なる。RS 232Cではコネクタの規格も統一されており、CRTターミナル・プリンタ・XYプロッタ・パーソナルコンピュータ等、このインターフェースを備えた機器システムとは簡単に接続でき、汎用性の高いものである。

4.3 PLCA (高速度システム間結合装置)

4.3.1 本システムにおけるPLCAの特徴

- (1) 16ビット転送を行っており、高速(最大450kB/s)である。
- (2) PFIシリーズの標準入出力制御プログラムを利用できる。
- (3) 標準的マイコン用周辺制御LSI 8255で簡単に制御できる。
- (4) ホスト・ターミナル双方のリモートIPLが可能である。
- (5) 非同期応答確認方式のため、相手装置の信号処理時間、線路の遅延時間を考慮する必要が無い。
- (6) 電源を共通にしたオープンコレクタドライバを使用し、送信の衝突におけるハードウェアの危険が無く、分岐接続が容易である。

4.3.2 インターフェースについて

図2に8255を用いたPLCAインターフェース例を示す。信号の種類を表1に、定義を表2に、本システムにおける転送制御手順を図3に示す。

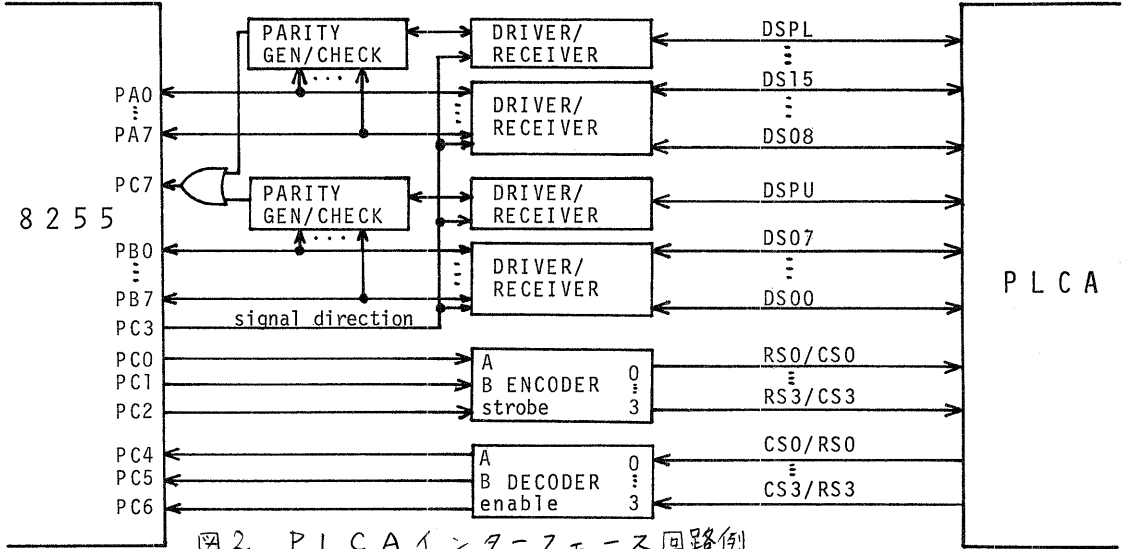


図2 PLCAインターフェース回路例

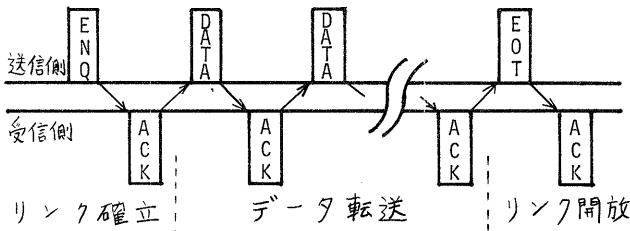


図3 PLCA転送制御手順

信号	動作
指令信号 CS	送信側から受信側に送られる制御信号
応答信号 RS	受信側から送信側に送られる制御信号
情報信号 DS	送受信双方で交換される情報信号

表2 信号の種類

	信号名称	機能
指令信号	CS0 受信開始指令(ENQ)	データ受信の開始を指示
	CS1 受信終了指令(EOT)	データ受信の終了を指示
信号	CS2 受信指令1(RT)	CS2, CS3は交互に送信されデータの受信タイミングを示し、受信データに関する応答を要求
	CS3 受信指令2(RT)	
応答信号	RS0 否定応答(NAK)	CS0~CS3に対する否定応答
	RS1 中断応答(RYJ)	受信装置から以後の送信の停止を要求
	RS2 肯定応答1(ACK)	RS2, RS3は交互に送信されCS0~CS3に対する肯定応答を示し、次の指令の送信を要求
	RS3 肯定応答2(ACK)	
情報信号	DS00 ~ DS15 情報信号	16ビットのデータ信号でCS2(3)と同時に送信される時は通常のテキスト, その他のCS/RSと同時に送信される時はCS/RSの修飾情報(注)
	DSPU	DS00~07に対応する奇数パリティ信号
	DSPL	DS08~15に対応する奇数パリティ信号

(注) 分岐接続時のターミナルアドレス識別, リモートコントロール信号の識別等。

表1 信号の定義

5. ターミナルの制御方式

ここでは、ターミナルのプログラムをどのようにローディングし、実行させるかを中心に、ターミナルの制御方式を分類する。

5.1 ROMによる固定プログラム方式

ROMに制御プログラムを格納しておき、通信回線はデータ転送だけに用いる(図4(a))。

5.2 ターミナルの補助記憶装置を用いてプログラムをローディングする方式

図4(b)において実行するプログラムを毎回補助記憶装置からローディングする方式であり、5.1よりは柔軟な制御を行える。しかし、実行すべきプログラムによって記憶媒体の選択など人手を要することもあり、また数少ない初期ローディングのために補助記憶装置を用意するのは不経済な場合もある。

5.3 ダウンローディング方式

補助記憶装置を持たないシステム、あるいは持っているも未知のプログラムを実行させる場合において、ホスト側から結合チャネルを通じてプログラムを送り込んで実行させる方式である。通常このダウンローディングは静的なもので、プログラム実行中は固定プログラムと同様に考えられる。

この方式はCCU(汎用通信制御装置)のように多種多様な複雑な処理を行う対して行われていることがある。

5.4 ダイナミックリモートローディング方式(DRL方式)

5.3の方式において実行中にもプログラムのローディングを可能としたのがこの方式である。この場合、ターミナルがホストに対してアップローディングを行うことも考えられるので、両方向ローディングをリモートローディングと呼ぶことにする。

さて、ここぞ問題となることは、ローディングの時期、通信回線上におけるプログラムとデータの区別である。そこで次の2つの方法が考えられる。

(1) プログラムをリモートモニタでロードする方法

ターミナル制御が一度ターゲットタスクよりリモートモニタ(Xメモリ等を操作できる基本モニタ)に移り、必要なリモートロードを行う方式である(図5)。

図中Cのリモートモニタスタートはターミナル側ターゲットタスクA終了時の命令で行うが、aのモニタスタートはホスト側からの割込みによって行う、リモートロードbはタスクAと並行処理可能であれば効率が良い。

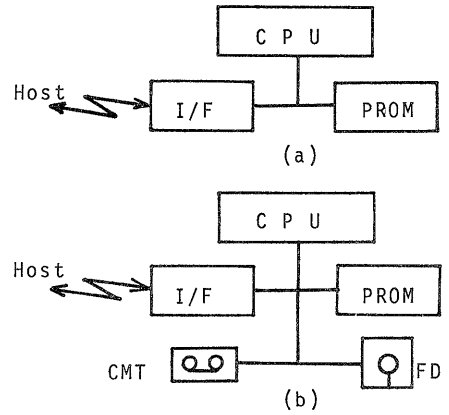


図4 ターミナルの基本的構成

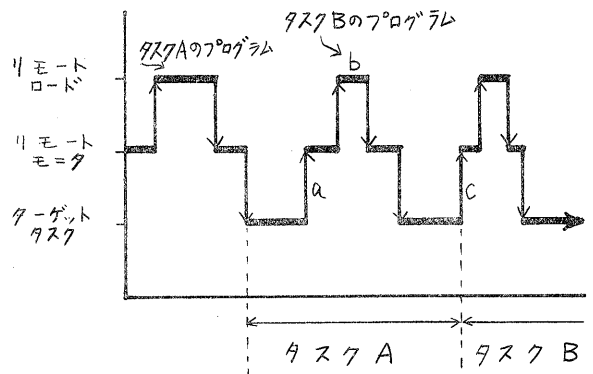


図5 プログラムロードのタイミング

(2) プログラムとデータを同等に扱う方法

この方法は、プログラムとデータが同形式の特別な場合に容易である。たとえば、LISPにおいては、プログラム(関数定義)とデータが同じリスト表現になっているため、一旦データとして受信したものをEVALし、プログラムとして実行することが出来る。一般のコンパイラ言語等では、プログラム実行中に呼び出せる、コンパイラ・リンカ・ローダ等が必要であろう。

6. システム運用と評価

実際にシステムを運用した結果、2.で述べた必要条件に対し表3のような評価が得られた。ここでは、特にチャンネル結合における転送形態を直列形と並列形に分けて考えたい。本システムのようにホスト側の標準的チャンネルを用い、ターミナルとしてマイクロ(パーソナル)コンピュータを結合する場合には、特に結合コストが問題であり、このとき両方式の特徴は表4のようになる。

このようにコストの面では直列形の方が良いと思われる。ただ、直列転送では無手順通信を行う場合、文字単位の処理となってしまいうため、あまり速度を上げられないことがある(ターミナルにマイクロコンピュータを用いたとき)。しかし、ターミナルのマイクロコンピュータに通信制御手順を付加するのは、それほど難しくなく、実際に3.4のPC-8001は無手順形では300 Baudの速度が最大であるが、簡易手順を付加することにより標準ハードウェアで最大の4800 Baudで転送可能となった。

それでも、数kB以上のデータ転送に存るとやはり直列形では待時間ができてしまう。そこで、本システムでは、U-1500・8086サブシステム間には、直列・並列両形式の結合を行い、目的によって使い分けしている。

項目	並列形	直列形
結合線数	多い	少ない
制御用LSI	少ない	誤り制御を含んだものも多い
統一規格	特に無い(単方向プリンタ用でセントロニクス型等)	RS232C(多くの装置・システムが持つ)等

表4 結合コストがらみた並列形と直列形の特徴

結合例(回)	付加ハード	距離	転送方式	結合目的	評価値				
					(1)	(2)	(3)	(4)	(5)
(1)	無	近	並列	ミニコンピュータ間汎用ファイル転送	×	◎	◎	○	○
(2)	無	近	直列	ミニコンピュータ間汎用メッセージ転送	△	×	○	○	○
(3)	有	中	直列	知能端末(アニメーション表示システム)	○	×	○	○	○
(4)	有	近	並列	非標準入出力装置制御	○	△	○	△	○
(5)	無	中	直列	マイクロプロセッサ開発援助システム	○	×	○	○	○
(6)	有	近	並列	(ダイナミック)リモートローディング	△	○	◎	○	○

(注) 2.計算機結合における必要条件による。記号は相対評価 X<△<○<◎

表3 計算機結合例の評価

7. 本システムの応用

ダイナミックリモートローディング(DRL)の機能を活用したシステムの応用として次のようなものが考えられる。

7.1 アクティブTSSターミナル

DRLを用いると補助記憶装置が無くてもターミナルの制御プログラムを動的に変更できる。ここではTSSのエディタにおける応用を考えてみたい。通常、TSSのスクリーンエディタは無駄なデータ転送を避けるべく、修正した部分のスクリーン内座標と修正内容だけを転送している。そのため、ホスト側では常にターミナルのスクリーン内容を把握している必要があり、わずかな行スクロール列スクロールを行っても再度スクリーン全体を転送している。ターミナル側で数スクリーン分のバッファによる管理を行えば良いが、表示属性の異なるファイルがある場合、ターミナルに全てのプログラムを持たせるのは不経済であり、DRL方式が適している。

また、エディタのSCAN機能(ソースプログラム入力時に行単位で各言語ごとの構文チェックを行う)等も、ホスト側で処理するよりDRLを行いターミナル側で処理する方が効率的であろう。

7.2 メッセージフロー実験システム

図6のようにメッセージ通信を行う並行処理タスクを、DRLによって各ターミナルに機能負荷を分散して割り当て、ターミナルのタスクを動的に変化させるものである。この場合、プログラムの転送が頻繁に起こるため、前述のごとく、並列(高速)転送と直列転送を使い分ける必要がある。

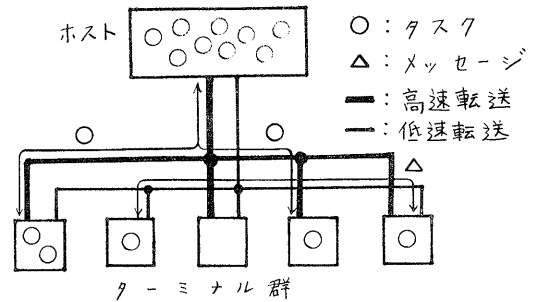


図6 メッセージフロー実験システム

8. あとがき

本システムは実験中のシステムであり、完成されたものではないが、いくつかのチャンネル結合によって以下のようなことがわかった。

- (1) 通常、マイクロコンピュータをターミナルに用いる場合、直列転送を行うと結合コストが小さくなる。
- (2) このとき、無手順ではなく、最小限応答確認の手順を設けることにより、かなりの高速化がはかれる。
- (3) DRL方式を用いれば、IPLのためだけに補助記憶装置を接続しなればなるなかつたターミナルシステムが、補助記憶装置を用いずに、その機能を果たすことが可能となる。

ここで提案したDRL方式は、一般の分散処理システムにおけるコンパクトなワークステーションをさらに高性能化する一手法であるとも考えられる。今後、種々の応用サブシステムを作成し、さらにDRL方式に検討を加える予定である。

[参考文献]

- (1) 横山, 小川, 田中 計算機間チャンネル結合方式についての一考察 情処全大21回 4G-7
- (2) 「PFU-1000シリーズ」ハードウェア通信装置編 富士通(株) 1980