Vol. 26 No. 3 情報処理学会論文誌 May 1985

図形処理端末のための分散処理方式の開発とその評価†

仁尾 都†† 片岡秀雄†††
越智利夫††† 竹屋弘史††††

設計から製造までを一貫して計算機処理する総合的な CAD/CAM システムを構築するには，大型データベースを擁する大型計算機の TSS システムと，それに回線接続するグラフィック端末を利用するのが，最も現実的な方法である．しかしながら，従来の TSS グラフィック端末はセグメント・バッファの内蔵により，通信回線速度の遅さをカバーするなど図形表示速度の高速化には十分注意が払われていたが，TSS の応答特性である入力に対する応答時間のばらつきや遅さなどの対策はほとんどなされていなかった．本研究の目的は，中央計算機の TSS 環境下でかつシングルタスクの制約下で稼動する応用ソフトに対しその入力に関する応答性向上を最重点においた分散処理方式を開発することにある．この目標を実現するために，端末に，入力プリミティブの集合であるコマンドの入力の全処理を行わせ，しかもコマンドの入力とその中央計算機側の応用処理とを，同時並行的に行わせる方式を実現した．この結果，中央計算機へのアテンション要求回数が 80%程度削減し，負荷分散に大きな効果があったこと，並行処理によって中央計算機側の応答時間のばらつきを小さくできたこと，および，コマンドの処理性能を 33% 向上したことを実験により確認した．

1. まえがき

中央計算機（以下ホストという）にグラフィック端末を最大 9600 BPS の通信回線によって TSS 接続すると大幅に応答性が劣化する．

そのため，最近，端末内に作画データ記憶メモリを所有させ回線接続による転送速度の遅さをカバーし表示性能を向上させる方法が一般化しつつある．また，3次元形状のワイアフレームモデルやソリッドモデルの処理を高速化するために，端末側に記憶する図形データの3次元化を行った例などもあるが，いずれも表示高速化を狙ったものである．一方，入力機能に関しては，グラフィックサブルーチンパッケージの国際標準である GKS や CORE において，点座標や文字列を入力の最小単位(入力プリミティブ)とし，その入力仕様を標準化することだけが行われていたり[1),2)]，さらに入力デバイスのよりいっそうの抽象化を狙った試みが行われているのみであり[3),4)]，入力応答の高速化を目的とした端末の高機能化については十分な研究が行われていなかったといえよう．

そこで筆者らは，入力プリミティブの集合であるコマンドを端末の最小入力単位とし，コマンドの入力とそのコマンドに先立って入力されたコマンドのホスト側の処理を同時並行的に実行する分散処理方式を開発した．

† A Method for Distributed Processing in a Graphic Terminal System by MISATO NIO (Hitachi Research Laboratory, Hitachi Ltd.) HIDEO KATAOKA, TOSHIO OCHI (Software Works, Hitachi Ltd.) and HIROSHI TAKEYA (Asahi Works, Hitachi Ltd.).

†† 日立製作所日立研究所
††† 日立製作所ソフトウェア工場
†††† 日立製作所旭工場

2. 従来の非同期入力方式の問題点

TSS 端末の入力応答性を向上するには，端末にできるだけ大量のデータを先行入力させ，一方では，ホストでそれらの応用処理を同時並行的に施すことが効果的である．そのためには，GKS 等が規定する非同期入力方式を利用する方法がまず考えられる．

いま，かりに**図1**のように，現時点での応用処理中の入力プリミティブを a_i とし，入力処理中のプリミティブを a_k $(i<k)$ としよう．a_i の応用処理の結果が a_j $(i<j<k)$ に影響を与える場合（たとえば a_j の削除等が発生した場合），a_j は入力に利用された後，影響を受けたのであるから処理の順序に矛盾が起きる．したがって，このような場合があるように，a_{i+1}〜a_k の入力の正しさはつねには保証されず，応用ソフトはそれらの待行列を読み捨て（フラッシング）しなければならなくなる状況も発生する．これでは，操作者は安心して入力作業ができない．しかも非同期入力を用いると，入力ガイダンス表示や入力データチェックが入力時点で行えないという致命的な欠陥を避けられない．

もし，GKS 等が提案するもう一つの入力方式である同期入力方式を用いると，入力ガイダンス表示や入力データチェックは可能となるが，非同期の応用処理のためにマルチタスク化が必要となる．この場合，使用言語の制限がある等，一般 TSS ユーザには，応用ソ

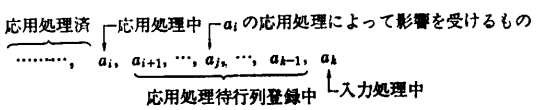


図 1 入力待行列中の入力プリミティブ
Fig. 1 Input primitives in input que.

フト作成上の，大きな制約になる．しかも，入力プリミティブ単位で入力タスクと応用処理タスクの受け渡しを行おうとすると，応用処理タスクの処理単位は小さくなる．また，入力タスクが応用処理タスクに次は何を入力すべきかを頻繁に問い合せることになる．結果的には，全体的に見ると同期処理とほとんど変わらなくなってしまう．以上のように，GKS 等の同期・非同期入力方式は，入力と応用処理の並行処理を行いたいユーザにとっては，不十分な方式であるといわざるをえない．

筆者らは，これらの欠点を克服するため，大量のデータを先行入力でき，かつ，その入力中にも，同時並行的に入力データを処理できる汎用的な入力制御方式があることを見いだした．その方式の特徴点は以下のとおりである．

① 応用ソフトが入力できる最小入力単位は，入力プリミティブとはせず，それらの集合であるとする．各集合の構成は応用ソフト開発者が定義でき，それぞれに独自の機能を割りつけることができる．端末は集合（以下コマンドという）単位に入力を実行する．

② コマンド単位に，そのコマンドの入力が完了後，続けて他のコマンドの入力を許すかどうか等の属性を，ユーザに与えさせる．

③ 応用ソフトには，コマンド単位の非同期入力方式を許す．

以降，本方式について説明をする．

3. コマンドの非同期端末分散入力方式

3.1 コマンドとメニューデータの設定

本端末システムでは，データ入力，ホストへの転送・応用ソフトによる処理など，すべての処理は，コマンド単位で行われるとした．コマンドの形式は以下のようである．

コマンド＝コマンド名＋パラメータ１＋パラメータ２＋…＋パラメータ n

パラメータの種類としては次の３種を設定した．

○C 種パラメータ：文字パラメータ（キーボード）
○P 種パラメータ：点座標パラメータ（ロケータ）
○E 種パラメータ：図形要素パラメータ（ピック）

応用ソフト開発者は，彼の用いるコマンド一つごとに，そのパラメータの個数や種類と入力案内文，E 種のパラメータごとの検索範囲指定情報，さらにはそのコマンドが入力された直後に，端末が施すべき前処理内容等を，本端末システムの設定したコマンド定義言語によって自由に定義できる．これらの記述文全体をメニューデータとよぶ．端末は応用ソフトごとに対応するメニューデータを記憶し，操作者が次々と入力するデータをコマンドに編集する．そのため，ホストの負担は大幅に軽減され入力応答も向上する．

なお，メニューデータの具体的な形式や応用ソフト開発者指定可能な前処理内容については 3.3 節で述べる．

3.2 コマンドの処理内容に応じた非同期処理の実現

個々のコマンドの処理内容に応じた効率のよい非同期処理を実現するために，各コマンドに以下の属性をコマンド定義言語により与えることができるようにした．

（1） ホスト転送属性：本属性が与えられたコマンドは入力が完了すると，ホストに転送され応用処理を受ける．もし，端末の前処理だけで済むような簡単なコマンドの場合は，本属性を与える必要がないが，この場合は，後で説明する前コマンド処理待属性が与えられてさえなければ，すぐに端末による前処理が行われ，引き続き次のコマンドの入力を許すことができる．すなわち，そのときにホストの応用処理が続行中である場合は，並行入力処理が実現することになる．

（2） ホスト入力指示待属性：本属性をもつコマンドの入力が完了すると，端末のコマンド入力続行は禁じられる．このコマンドを処理する応用ソフトが発行する入力指示待解除指令が端末に到着すると入力は再開する．ユーザは，応用処理で端末に記憶中の作画データを削除したり，その表示位置の変更をしたりする指令を端末に発行する可能性のあるコマンドに対しては，本属性を与えなければならない．逆に，作画データの追加等，他コマンドの先行入力の結果に影響の及ぼす可能性のないコマンドには本属性を与えるべきではなく，このようにすればこのコマンド入力直後も他コマンド入力の続行が可能となり，応用処理と並行して入力処理ができる．

（3） 前コマンド処理終了待属性：この属性をもつ

コマンドの入力が完了すると，端末はそれ以前に入力された全コマンドの応用処理が完了したことを確認後，はじめてこのコマンドの前処理を行う．たとえば，端末内の作画データ格納エリアの空きサイズなどの端末状態問合せを端末の前処理で行わす場合はこの指定をしなければならない．

（4） ホスト処理結果返送属性：ユーザは，応用処理の結果，なんらかの指令を端末に送ることになるコマンドに対しては，必ず本属性を与えなければならず，かつそのコマンドの応用処理終了時には端末に対し終了宣言を行う義務をもつ．これにより，端末はどのコマンドが現在応用処理進行中であるかをつねに把握することができ，また，前コマンド処理終了状態も知ることができる．

3.3 メニューデータの構造

端末独自で，コマンドの入力をその前処理およびホストとの同期連絡を行うことができるように，応用ソフト開発者の定義が可能なメニューデータを端末に記憶させる機能を与えたことはすでにのべた．本端末システムでは，タブレットを用いた紙メニュー方式をとっており，**図2**に示すようにタブレット上に紙メニューを置く領域を設け，ここからコマンド名やC種パラメータの入力をすることとした．さらにタブレット上にディスプレイ管面と一対一に対応させる領域を設定し，ここをP種およびE種のパラメータ入力に用いた．これらの入力制御手順を記述したメニューデータのデータ形式を**図3**に示す．

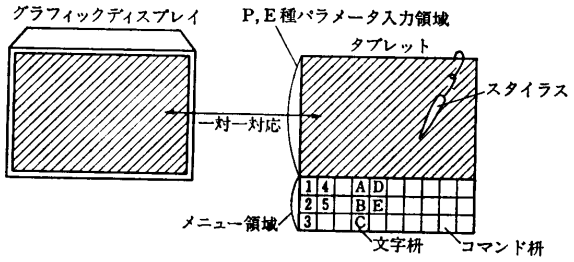


図2 タブレットの利用方法
Fig. 2 Role of a tablet.

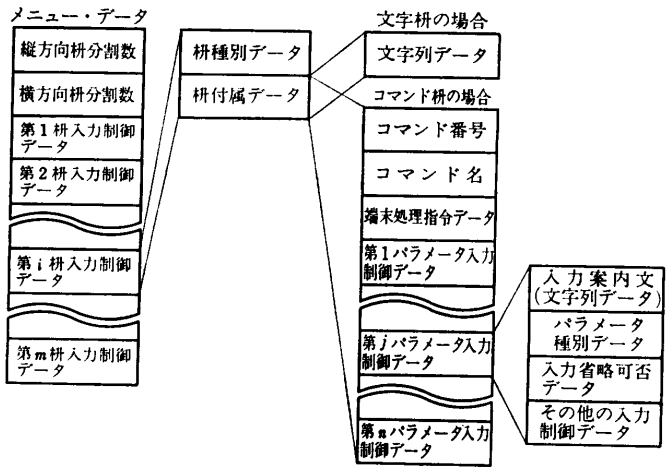
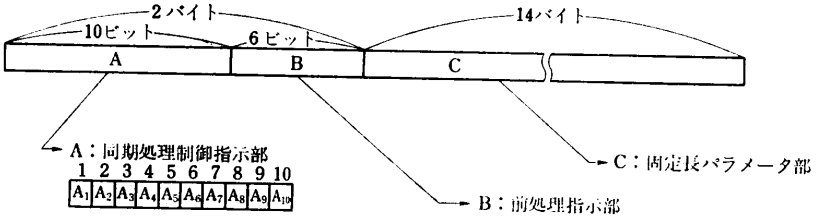


図3 メニューデータのデータ形式
Fig. 3 Data format of menu data.

メニューデータの先頭部は，メニュー領域が何個のマスに分割されているかを示し，さらにおのおのはコマンドを入力するためのマスまたはたんに文字を入力



A_1：サブコマンドフラグ
A_2：予備
A_3：ホスト転送フラグ
A_4：ホスト処理結果返送フラグ
A_5：ホスト入力指示待フラグ
A_6：ホスト表示指示待フラグ
A_7：予備
A_8：前コマンド処理終了待フラグ
A_9：リフレッシュフラグ
A_{10}：予備

0：無処理
1：アクティブレベルセット
2：CRT枠退避
3：CRT枠復帰
4：層表示
5：層非表示
6：削除
7：平行移動
8：x方向移動
9：y方向移動
10：任意図形反転
11：任意図形回転
12：定形図形反転
13：定形図形回転
14：図形コピー
15：紙テープ読込
16：フロッピディスク読込
17：特定定形図位置読込
18：全定形図位置読込

図4 端末処理指令データ
Fig. 4 Control data for local processing.

表 1 メニューデータ例
Table 1 Example of menu data.

コマンド名	第1パラメータ		第2パラメータ		ホスト転送	ホスト入力指示待
	案内文	種類	案内文	種類		
抵抗配置	配置位置は？	P	−	−	○	×
素子名	素子は？	E	名前は？	C	○	×
結線	始点端子は？	E	終点端子は？	E	○	○

するためのマスかの区別を記憶している．コマンドマスであれば，さらにそのマスの属性としての，コマンド名やそのパラメータの入力案内文，パラメータ種類情報等が付加される．

以上はコマンド入力のための制御情報であるが，コマンドが入力された後に端末が前処理すべき内容を記憶した端末処理指令データもメニューデータの中に，各コマンドごとに与えられている．端末処理指令データの具体的形式は**図 4** に示す．この先頭部 A は，3.2 節で述べたコマンド同期処理属性を記憶する部分であり，前処理指令部 B は端末のなすべき前処理内容を示す．ユーザはここに示す以外の内容を端末で実行指示することはできず，その場合はユーザの作成するホスト側の応用ソフトで行わなければならない．固定長パラメータ部 C には，前処理の対象となる図形要素名やE種パラメータの検索範囲指定情報が記憶される．**表 1** に，本端末の一つの応用ソフトであるシーケンス CAD システムで実際に定義されているコマンドの一部の例を記載した．同表中の抵抗配置あるいは素子名指定コマンドに対する応用処理では，素子形状作画データやテキストデータを端末に転送するのみであり，削除コマンドを発行しないので，ホスト入力指示待属性が与えられていない．一方，結線コマンドの場合はすでに配線されている1本の線から，T 字型に分岐配線しようとすると，いったん，その線を削除し，代りに2分割した線を再転送して表示する必要があるため，削除コマンドを発行する事態が生じる．したがって本コマンドに対してはホスト入力指示待属性が与えられている．いま，抵抗配置コマンドが複数連続投入された後，結線コマンドが投入されて，端末が入力禁止状態になったと仮定しよう．シーケンス CAD システムは，その時点ではまだ，端末が入力を中止したことは連絡されず，相変わらず，すでに先行的に入力されて待行列に登録されているコマンドを一つずつ取り出しては応用処理を施していく．ついに最後に入力した結線コマンドの応用処理を開始したときに，初めて端末が入力禁止状態にあることを知る．そこでその応用ソフトは，削除指令等を発行後に，ホスト入力指示待解除指令を発行し，これを受けて端末は入力を再開する．したがって，操作者は，どのコマンドはどのような同期処理属性をもっているかを知っている必要がなく，また，いったん入力の完了したコマンドが後になって読み捨て（フラッシング）されることもないので安心して任意のコマンドを連続投入できる．

3.4 コマンドの同期・非同期入力

GKS 等は，入力プリミティブに対して同期・非同期入力の2方式を提唱している．本システムでは，これに対応するものとして，コマンドに対して同期・非同期入力の2方式を実現した．3.3 節で説明したような，待行列に入力コマンドを次々に登録し，応用ソフトが行列から一つずつ，ファーストイン・ファーストアウトでコマンド処理する方式はコマンドの非同期入力処理というべきものである．本システムでは，さらに，一つのコマンドの入力が完了したら入力禁止状態になり，その応用処理が終わり，応用ソフトが次のコマンド入力を要求した段階で初めて端末が入力が再開されるというコマンドの同期入力処理方式も可能とした．この同期はホスト処理結果返送属性を利用して実現される．

両方式の選択は応用ソフトの選択にまかせず，オペレータが入力する本端末システム固有の基本コマンドにより，端末の入力モードを切り換えることができるようにした．これにより，操作不慣れな操作者は，同

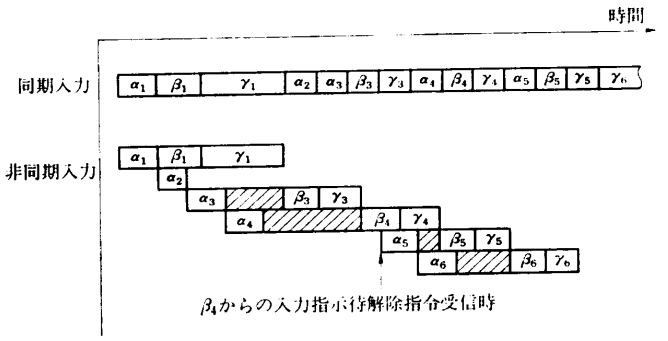


図 5 コマンド単位の同期・非同期入力処理手順
Fig. 5 Parallel processing procedure of commands.

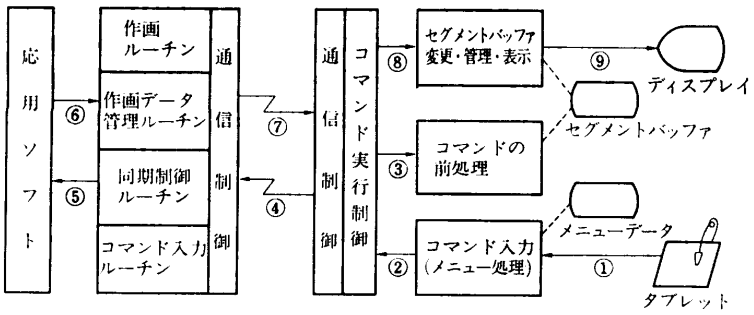


図 6 ソフトウェアのシステム構成図
Fig. 6 Configuration of software system.

期入力機能を用いて，逐一結果を確認しながら次のコマンドを入力でき，熟練の操作者は非同期入力により高速オペレーションができるなど，任意時点での自由なモード切換えが可能となる．

図 5 は，本端末システムにおけるコマンドの同期・非同期入力処理手順の相異を示したものである[5]．α_i は第 i 番目のコマンドの入力処理を，β_i は応用処理を，γ_i は応用処理結果の転送処理ならびに端末の結果表示処理を示す．本例では，2番目以外のコマンドはすべてホスト転送属性が与えられており，4番目のコマンドだけにホスト入力指示待属性が与えられている．したがって，非同期入力モードでは，$\alpha_2, \alpha_3, \alpha_4$ は直前のコマンド入力の完了直後にホストの応用処理と並行して開始されており，α_5 は，β_4 の応用処理の途中で発行される入力指示待解除指令が端末に到着した段階で開始される状況がこの図に表現されている．

3.5 システム全体図

本端末システムにおける処理の流れの概略を**図 6** に示す[6]．コマンド入力プログラムは，オペレータが次々と入力するデータをメニューデータが指示する処理手順にしたがって整理・編集し，コマンドを生成する(①)．1コマンドの入力が完了したら端末はメニューデータに記載してあるそのコマンドの前処理を実行する(②，③)．さらに，ホスト転送属性が与えられていれば，端末内のコマンド転送待行列バッファに登録する．さらに，ホスト入力指示待属性が与えられているかを調べ，与えられているならば端末を入力禁止状態にし，与えられていなければ，次のコマンドを入力するようオペレータに催促する．端末は，上記の処理を行う一方で，マルチタスクにより，コマンド転送待行列バッファに登録されている複数コマンドを，ホスト側の基本ソフトに一括転送する処理を行う(④)．ホスト側の基本ソフトは，受信したコマンド群のなかから一つずつ応用ソフトに渡し，応用処理をさせる(⑤)．応用ソフトは処理の結果(たとえば，作画データ追加指令，削除指令，あるいは同期制御指令など)を端末に転送する(⑥，⑦)．端末は受信したこれらの指令をセグメントバッファ内の作画データに反映させ，ディスプレイに表示する(⑧，⑨)．

図 7 は本端末システムのハードウェア構成図である．ディスプレイは蓄積管形とリフレッシュ形（ラスタスキャン方式）の利用を可能とし，目的によって使い分ける．**図 8** は端末外観図である．

4. 性能評価

本端末システムの一つの応用ソフトであるシーケンス CAD システム[8] を用いて，**図 9** の回路図を各モード下で入力したときの入力所要時間の測定結果を**表 2** に示す．表2の時間は，1人の操作者が図9の原図を参照しながら，まずタブレット上の紙メニュー

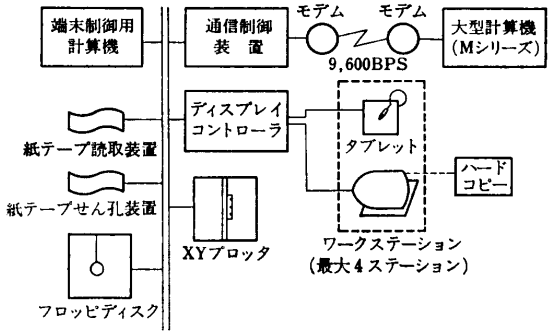


図 7 ハードウェア構成
Fig. 7 Hardware configuration.

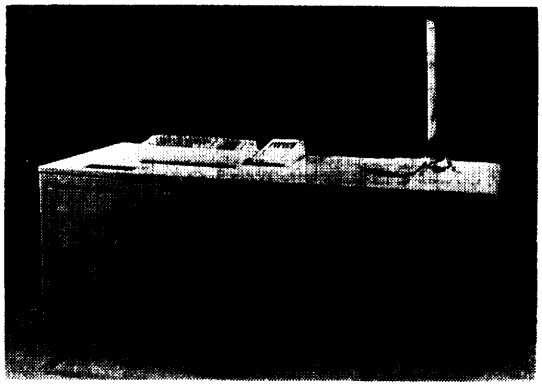


図 8 インテリジェント端末
Fig. 8 The intelligent terminal.

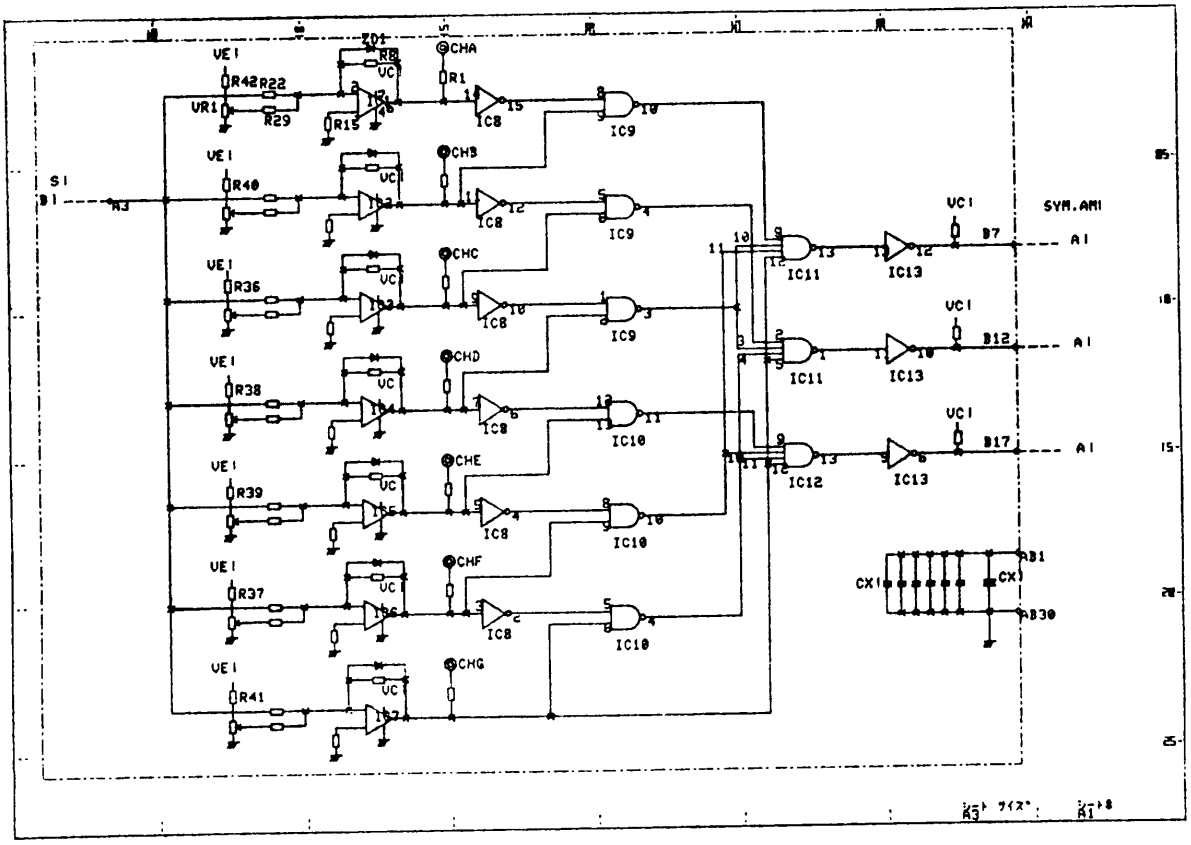


図 9　論理j図
Fig. 9　Circuit diagram.

表 2　各モード下の図面入力所要時間
Table 2　Input time of drawings.

実験番号	モード	
	同期入力（秒）	非同期入力（秒）
1	2,987	2,384
2	2,902	1,954
3	2,075	1,741
平均	2,655	2,026

を用いて素子の配置コマンドを入力するところから始まって，結線を経て素子名を入力することにより，全図面をディスプレイ上に完成させるまでの通し時間である．計6回の実験を通じて，操作者は一名に限定し，まったく同一の操作，すなわちコマンド入力の順序を守らせた．測定は，コマンドの同期入力モード，非同期入力モード下で各3回，計6回行った．この結果，同期入力モードに対し，非同期入力モードでは，24% の入力時間の短縮効果があった．なお，本実験では，入力コマンド数は158でタッチ数は833であった．ホストに対するアテンション回数は 80% が削減されることが確認できた．1アテンション当り，0.5秒だけ余計にアテンション応答所要時間が短縮されたと仮定すると，本端末利用による平均図面入力時間の短縮効果は417秒と予想される．結局，最大，約33%の入力時間が削減できたことになる．1タッチとは，1回の入力機器操作を意味し，たとえば，タブレットからの一つの点座標入力操作あるいは，キーボードからの一つの文字列入力操作をいう．一般の端末では，1タッチごとにホストにアテンションが発生する．表2からわかるように，同一実験を3回くりかえしたが，そのたびに大きな時間短縮効果が見られたのは，操作者がしだいに操作に慣れてきたためであると思われる．

全入力時間の短縮効果のほか，応答時間のばらつきが平滑化される効果も大きい．TSS は一般に応答時間のばらつきが大きい．本端末は，コマンドパラメータの入力までも端末で行っているので，コマンド入力中の応答のばらつきはない．さらにこれを一歩進めて，コマンド入力後の，そのコマンドの中央計算機による応用処理に対する応答時間のばらつきをさらに削減しようとするものが非同期入力モードである．同表

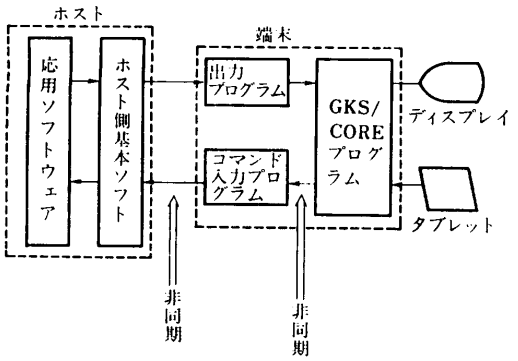


図 10 本端末の入力機能の GKS による実現方式
Fig. 10 Terminal's input program on GKS.

から，同期入力モードにおける図面入力時間のばらつきが改善されているのが読みとれる．

5. GKS, CORE との関係の考察

本研究のコマンド入力に関する端末機能が，GKS や CORE の提案する入力インタフェースと，どのような関係にあるかを，以下に考察する．

本端末システムは，コマンドないしパラメータ入力中に，いつでもシンボルコマンド（タブレット上で特定記号や図形を手書き入力することにより，計算機に意味を認識させることによって与える指令）や種々のコマンド入力制御指令（パラメータ区切り指令，パラメータ入力省略指令）などが入力可能である．したがって，本質的に非同期入力を行いつつも，オペレータの誤操作防止のために，データチェックを行い，結果的にはある程度の同期的入力機能を果たしているといえよう．

以上の理由から，コマンド入力機能は，GKS 等からみると，一つの応用ソフトである．すなわち，**図 10** のように，そのコマンド入力のための応用ソフトは端末にローディングされており，それは，GKS 等の非同期入力機能を使用しているとみなせる．

一方，本端末システムは，さらに，応用システムに対して，コマンド単位の非同期入力機能を支援している．GKS 等にとっては，この機能は，本端末システム独自のホスト側基本ソフトと，端末側のコマンド入力プログラムという，二つの応用ソフトがマルチタスク化によって実現する応用機能であるとみなせる．

6. おわりに

図形処理分野における，ホスト側と端末間の一つの分散処理方式として，端末内にコマンド入力用のメニューデータを格納することにより，端末にコマンド単位の入力を，また，ホストにそのコマンドの応用処理を行わせ，しかも，両処理を同時並行的に行うコマンド処理能力の高い分散処理方式を開発した．

本方式は，GKS や CORE が提案する入力プリミティブを入力および処理の単位とする方式に比べ，中央計算機側の入力処理の負荷を 1/5 に軽減し，図面入力時間を 2/3 にした．また，ホストの処理に対する応答待時間の平滑化や，応用ソフトに対する入力プログラム開発負担の軽減，さらには，各種応用ソフト間でコマンド入力操作の共通化が可能となった．

今後の課題としては，

（1） 端末の現在のコマンド入力機能に対し，マクロコマンドの定義とその展開機能を追加する．

（2） 一つの応用ソフトをホスト側と端末側とに分割ローディングし連動させる方式を開発し，応答性の向上を図る．

ことがあげられる．

謝辞 本研究は，日立製作所の図形処理端末システム GRADAS の開発にそって行われたものである．この間，指導・協力いただいた日立製作所の徳増，国友，矢嶋，星野，大木，三島の各氏，および日立エンジニアリングの森実氏他，関係者の方々に厚く感謝する．

参考文献

1) Boro, P. R.: GKS the First Graphics Standard, IEEE CG & A, pp. 9-23 (July 1982).
2) Status Report of the Graphics Standards, Planning Committee, Computer Graphics (August 1979).
3) Hamlin, G.: Software for Device-Independent Graphical Input, Graphics Interface '82, pp. 23-27 (1982).
4) Borufka, H. G.: Dialogue Cells, a Method for Defining Interactions, IEEE CG & A (July 1982).
5) 仁尾 都：図形処理端末処理における中央計算機との平行分散処理方式，情報処理学会第 21 回全国大会講演論文集，pp. 1157-1158 (1979).
6) Nio, M.: Distributed Processing Terminal System for CAD/CAM, COMPSAC '79—IEEE Computer Societies Third International Conference (1979).
7) Tokumasu, S.: Implementation of Geometric Modelling System—HICAD, Intergraphics '83 (1983).
8) 小林克己：制御盤，配電盤，整流器盤における CAD/CAM システム，日立評論，Vol. 62, No. 7, pp. 5-10 (1980).

（昭和 58 年 10 月 4 日受付）
（昭和 59 年 12 月 20 日採録）