1N-9 科学技術計算用データ駆動計算機 ＳＩＧＭＡ－１における入出力の実現

平木 敬　関口智嗣　島田俊夫
(電子技術総合研究所)

１．はじめに

現在当所で研究・開発を進めている科学技術計算用データ駆動計算機ＳＩＧＭＡ－１は、データ駆動計算機の科学技術計算への適合性を実証することを目的としている。実用規模の数値計算プログラムは大量の入出力を伴うことから、この目的を達成するためには効率的入出力の実現が不可欠である。しかしながら、ＳＩＧＭＡ－１は並列処理を行なうデータ駆動計算機であり、入出力の実現方式はフォンノイマン型単一計算機と大きく異ならざるを得ない。

逐次型言語における入出力は、プログラムの逐次的解釈により、入出力データの構造が解釈される。それを並列処理計算機にそのまま導入すると、入出力操作がプログラム全体の逐次化を引き起こし実行効率が低下する。

問題の持つ並列性を生かし、実行効率の低下を最小限に止めるためには、逐次化が本質的でない入出力操作は並列に行なう方式に拡張することが必要である。

一方、フォンノイマン計算機では、入出力操作は割り込み処理と深く関係し、多大のオーバーヘッドを産出してきた。データ駆動計算機では、全命令が同期操作を兼ね備えていることから、外界と計算機の内部処理を容易に結合することが可能である特長を持っている。このことは、特に信号処理装置を作製する場合に大きな福音となる。

ＳＩＧＭＡ－１では、データ駆動方式の特長を生かし、かつ逐次言語との連続性を満たすため、３種類の入出力形態に分類して入出力操作を与えた。本発表では、ＳＩＧＭＡ－１における実現形態について述べる。

２．入出力形態

ユーザにとって入出力とは何であるかは、計算すべき応用分野、記述言語、解釈実行の基本方式（フォンノイマンかデータ駆動か）により様々な解釈が可能である：

(1) 外界、特に人間との対話的入出力。キーボードからのストリーム入力やディスプレイへの出力が代表的なものである。
(2) 外界からの信号入出力。信号入出力という特徴は、その構造が先験的に定まっていることに集約される。
(3) 拡張された記憶としての入出力。数値処理における初期データ入力、中間結果入出力、結果出力等が該当する。この場合は、計算機内部のデータ構造を、記号的に外部のファイルと結合して入出力が行なわれる。

逐次処理計算機では、一時に一個のプログラム部分しか実行しないという本質的特性から、これら全ての入出力をストリームという概念で統一的に扱うことが可能である。実際、Ｃ言語等では言語自身は入出力機能を持たず、入出力ストリームを解釈するサブプログラムにより統一的に入出力を扱っている。しかしながら、プログラムを並列に実行する環境では、ストリームによる統一的解釈を実現するために全域的な全順序化が必要でありる。

３．ＳＩＧＭＡ－１の入出力環境

ＳＩＧＭＡ－１はバックエンド計算機として使用することを前提として設計されている。従って、入出力操作も、ホスト計算機と協調して動作を行なう。演算処理装置は命令レベルのデータ駆動計算機、構造体処理装置はベクトル等構造体に対するアクセス、領域管理を行ない、メンテナンスプロセッサに接続するチャネルを内蔵する。チャネルはメンテナンスプロセッサからは構造体処理装置をアクセスするパケットインターフェイスとして動作し、Ｂ－構造体の非同期読みだし、書き込み、構造体の割り付け、構造体に全てデータが格納されたことの検査を行なう。　ホスト計算機は、ＳＩＧＭＡ－１フロントエンドシステムの中心部分を担当するワークステーションであり、ネットワークを介して演算処理装置、構造体処理装置と結合しているとともに、メンテナンスプロセッサと高速に交信することが可能である。

４．入出力操作の実現

４－１．対話的入出力

逐次処理言語で記述されているプログラムでは入出力プリミティブはサブプログラムに分散的に配置されている。プログラムの実行に伴い、実行される入出力プリミティブの順番でストリーム入出力を行なう。これをそのまま並列環境に持ち込むと、並列に動作するプログラム要素はデータ依存関係で決定する半順序関係でしか実行順序が規定されないため、入出力プリミティブの実行順序は一般には規定されず、各プリミティブがどの要素に対して操作を行なうかは決定不可能である。

ＳＩＧＭＡ－１ではストリーム入出力に専用命令（ＧＥＴおよびＰＵＴ）を用意し、プログラム中の全プリミティブに全順序関係を持ち込む方式を採用している。ストリーム入力の概要を図１に示す。ＧＥＴ命令はストリーム識別子を入力とし、入力システムコールとストリーム識別子を出力とする。ストリーム識別子は、プログラムで用いられるストリーム毎に１個生成され、ストリームの識別と、ＧＥＴ命令間の全順序を規定するために用いられる。ＧＥＴ

Input and Output System for the Dataflow Supercomputer SIGMA-1
Kei HIRAKI, Satoshi SEKIGUCHI, Toshio SHIMADA
Electrotechnical Laboratory

命令は更に、ストリーム識別子をインクリメントし次のＧＥＴ命令へ送る。

入力システムコール本体では、ストリーム識別子のカウンタフィールドを検査し、入力の全順序関係を保存する操作を行なう。当該入力の処理はホスト計算機にＩＮＣＡＬＬ命令を用いて入力要求を行なう。ストリームのデータは、ＳＩＧＭＡ－１とパケット・インターフェイスを介して結合しているホスト上に存在し、ＩＮＣＡＬＬをグローバルネットワーク経由で受け取ると、次のデータ要素を演算処理装置に送り、入力動作を実現する。ストリームとデバイスの結合、物理的入出力操作は全てホスト計算機が実行し、ＳＩＧＭＡ－１上のプログラムとの通信は全てＨＯＳＴＣＡＬＬ命令を介して行なわれる。

ＧＥＴ命令が条件分岐構文中に存在する場合は、ストリーム識別子にも条件スイッチをかける。

ＰＵＴ命令は、出力データを入力オペランドとして取る点を除き、ＧＥＴ命令と同一の処理を行なう。

対話的入出力の本実現方式はオーバーヘッドが大きいが、逐次プログラムにおける入出力をシミュレートする能力を持つため、広い応用範囲を持つ。

４－２．信号入力

信号入出力はパケットアーキテクチャを用いるデータ駆動計算機の利点が生かせる入出力形態である。すなわち、入力アークが開放されている命令は全て外界からの入力を待っていると見做すことが可能である。信号出力は同様、パケットインターフェイスを出力アドレスとして、全ての命令で実現可能である。従って、入出力のための特別の命令実行は全く存在せず、オーバーヘッドはない。

４－３．ファイル入出力

ＳＩＧＭＡ－１では拡張された記憶としての入出力はファイルに対する入出力として実現される。

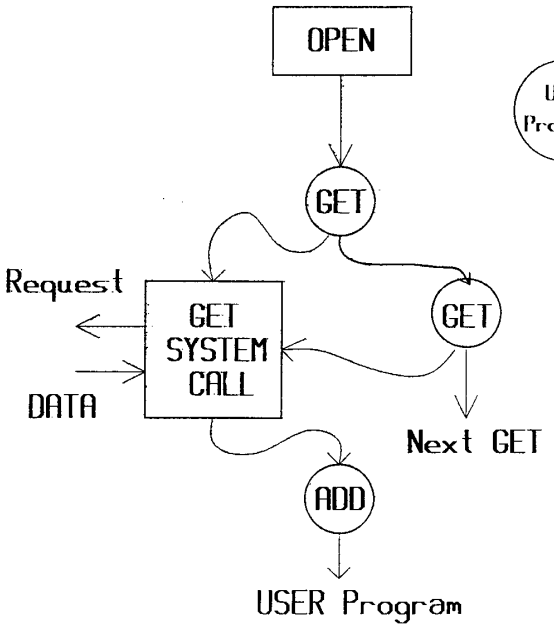

図１．ストリーム入力

前記２種類の入出力が、データ駆動方式特有のパケット転送により実現していることと比べ、ファイル入出力は構造体をバッファとして使用する通常のブロック入出力の形態を取る。

入力は、ブロック入力要求をサービスプロセッサへパケットを経由して通知すると、メンテナンスプロセッサからＩ／Ｏチャネルを経由してバッファにディスクの内容が転送される。バッファとして使用するＢ－構造体は同期のためのフラグを持っているため、転送が終了しないうちに読みだし要求を受理することが可能である（図２）。

バッファに対する出力は、構造体処理装置で転送語数を計数することにより、ブロックの書き込み終了を検出し、サービスプロセッサへディスクへの転送を起動する。出力の一貫性を保つため、計数はユーザプログラムとバッファにおいて二重に行ない、プログラム終了時には両者の一致を待ってブロックの掃き出しを行なう必要がある。

５．おわりに

上記のように、ＳＩＧＭＡ－１では３種類の入出力方式を目的に応じて組合せ、幅広い応用プログラムを効率良く実行できる設計を行なった。しかし、より広範囲な入出力を効率的に実現するためには、現在のＳＩＧＭＡ－１の機能に加えて、① 多次元入出力識別子、② 並列識別子（所謂カラー）とデータの混合演算、③ 構造体アドレスを入出力識別子から直接生成する能力、④入出力の効率化に必要なマッチング・ファンクション、⑤入出力バッファに用いる構造体のハードウェアによる管理をアーキテクチャとして取り込むことが望まれる。

なお、本研究は大型プロジェクト「科学技術用高速計算システム」の一環として行なった。本研究を支えていただいた柏木寛次長、弓場敏嗣計算機方式研究室長に深く感謝いたします。

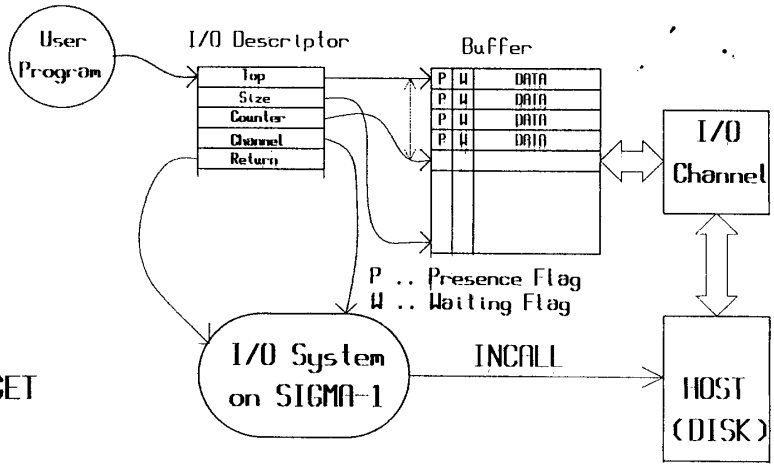

図２．ファイル入力概念図