

拡張性の高いシステムのバス結合方式と制御方式

A Bus Structure and Control for a Highly Expandable System.

—ポリプロセッサ・システム EPOS-2 のシステムバス—

The System Bus of Polyprocessor System : EPOS-2

山崎 勇, 前田 明, 宮田 操, 神谷 茂雄, 笠井 公
Isamu YAMAZAKI, Akira MAEDA, Misao MIYATA, Shigeo KAMIYA, Hiroshi KASAI
東京芝浦電気株式会社 総合研究所
TOSHIBA RESEARCH AND DEVELOPMENT CENTER

[1] はしがき

最近のLSI技術の進歩により、簡単な電子計算機のCPUは1個のLSIの上に実現することが可能となって来た。LSI化されたCPUは極めて小型であり、かつ同種多量生産により安価に製造できるため、これを多数用いて1台のより大きな計算機(ポリプロセッサ・システム)を構成することが考えられる。

1個の強力なCPUを備えた従来の大型計算機が少数精鋭主義の極限とすれば、ポリプロセッサ・システムは、多数の凡才を集めた大海戦術のようなものである。

このようなポリプロセッサ・システムでは、次のような可能性を持つことが期待される。

- (a) 同一ハードウェア、同一アーキテクチャで、小規模から大規模までのシステムを構成し得る可能性。
- (b) 一旦設置稼働後も、処理量の増大に応じて簡単な追加により処理能力を増大させ得る可能性。

しかし他方、ポリプロセッサ・システムには、次のような問題点もある。

- (x) ハードウェア構成の変化に応じられるオペレーティングシステム(OS)は複雑で、効率の悪いものになるおそれがある。
- (y) CPU間の競合や連絡のオーバーヘッド等により、CPUの個数に比例しては処理能力が

増加しないおそれがある。

本稿で「拡張性の高いシステム」とは、上記(a),(b)の可能性を実現し、(x),(y)の問題点を軽減したシステムを言うこととする。ポリプロセッサ・システムを拡張性の高いものとするためには、その結合方式と処理形態とをどのように選択するかが最も重要である。

我々は「PULCE」と呼ぶ、ゲート数7000以上のLSIプロセッサを演算部の中心に用いてポリプロセッサ・システムを開発している。そのプロトタイプ(EPOS-1)⁽¹⁾では上記(a),(b)に重点のある構成であった。EPOS-2は、これをさらに上記(x),(y)の点で改善したものである。その結果、ポリプロセッサ・システムの結合方式として最も適当な方式の一つと思われる方式(独立並設多バス方式)に到達したので、以下にその概要を報告する。

[2] 結合方式の検討

ポリプロセッサまたはマルチプロセッサにおける結合の対象(モジュール)としては、通常CPU、メモリ、およびI/Oプロセッサ(IOP)がある。これらを結合する形態としては大きく分けて、単一バス結合、メモリスイッチ結合、ランダム結合がある。⁽²⁾

単一バス結合は、図1に示すように、CPU、メモリ、IOPが単一のバスに結合したものであ

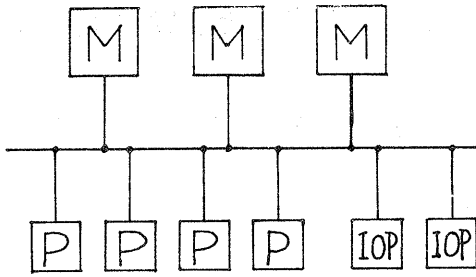


図1 単一バス結合

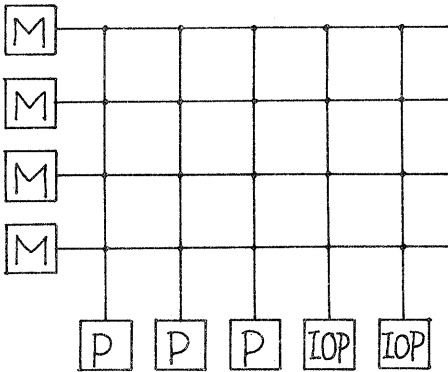


図2 メモリスイッチ結合

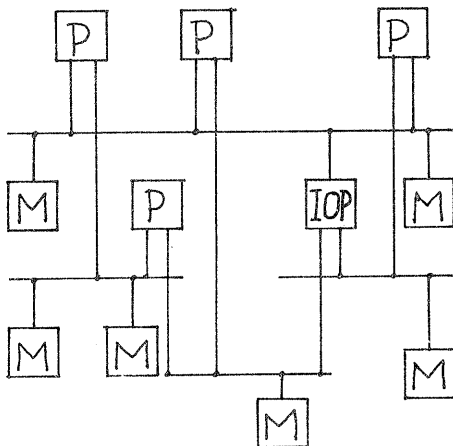


図3 ランダム結合

る。この方式では結合が単純で一次的であるために、最も経済的であり、各モジュールの追加・削除は技術的に比較的容易である。またその結合形態が斉一で規模の拡大が一次的であるため、OSの制御機構も特に複雑にならない。反面、バスは1本しかないので、全体の処理能力はこのバスの転送能力でおさえられ、CPUの数を増してもある限界数以上では処理能力が上がらなくなる。この点は、このバスを特に工夫して高速化し、限界数を実用上十分な程度に引き上げることが考えられるが、バスが高価となり小規模の構成ではかえって不経済となる。またいずれにしても、このバスに障害が起ると、システム全体が動作不能となるという大きな欠点がある。

メモリスイッチ結合は、図2に示すように、CPU群とIOP群とが、マトリクス状のメモリスイッチによりメモリ群と結合したものである。この形態では各CPUまたはIOPと各メモリモジュールとは、考え得る最大の転送路で結合されているため、この形態を保ったまま規模を拡大できれば、転送能力が処理能力のネックとなることはない。また通常このスイッチはプログラマから透明に作られ、各CPUと各メモリモジュールはプログラマから等価に見えるように作られるので、OSは複雑にはならない。また各交点のスイッチを独立に作ることによりスイッチの障害に対しても強くできる。しかし、この結合方式で規模が少し大きくなると、メモリスイッチのハードウェア量がかなり大きくなりこの部分が高価となる。さらにCPUの増加とメモリの増加の双方に応じて二次的に拡大し得るメモリスイッチは、回路的に複雑になる。

ランダム結合は、CPUとメモリの結合関係が一様でないような結合である。図3はその一例である。この例ではCPUは複数のポートを有し、システム内のいくつかのバスに自由に接続されている。メモリモジュールとIOPもそれぞれのバスのいずれかに接続される。この形態は結合関係が自由であるので、アプリケーションが限定されればそれに応じた構成を取ることが比較的容易である。またモジュールの追加もハードウェア的には容易にできる。しかし、結合関係に斉一性がなく、いわば多次元的に拡張され、プログラマから透明でもないため、OSはかなり

複雑となり、どの構成でも共通に使えるOSを作ることは非常に困難である。

このように、経済性では単一バスのような形態が有利であり、転送能力ではメモリスイッチのような形態が適しており、OSを複雑にしないためには結合が音一でプログラマから透明であることが望ましい。

これらを考慮して、EPOS-2では多バス方式の一種を採用した。これは単一バスの持つ経済性、音一性、透明性を生かして、メモリスイッチの持つ転送能力の拡張性を取り入れたものと言える。

S-BUSと呼ぶEPOS-2のこの結合バスは、一言で言えば、互いに独立な単一バスを複数本並行に設けたものである。その本数は可変であり、またどのバスが用いられるかはプログラマからは見えない。そこでこの方式はいわば、独立並設多バス方式と言える。図4はその結合形態を示したものである。

表1にS-BUSと他の3方式の比較を示した。S-BUSは複数のバスのどれを使用するかを決定する機構が必要のため、単一バスよりは高価となる。しかしその他の点では望ましい特性を備えている。ここで改めて、S-BUSの利点を要約すれば、次のとおりである。

- (1) 本数の増減により転送能力を増減できる。
- (2) 各バスは互いに独立であるため、一部が障害を起しても、残りのバスを用いてシステムは動作をつづけることができる。
- (3) 使用バスの選択はハードウェアで行ないプログラマは意識しなくてよく、また規模の拡大の際にも音一性が保たれるため、OSは複雑にならない。
- (4) 比較的経済的である。

[3] EPOS-2の構成

ポリプロセッサシステムEPOS-2の概略構成を図5に示す。

本システムはS-BUSを中心として、多数のコンピュータモジュール(CM)と、入出力制御モジュール(IOM)と、共用メモリモジュール(SMM)から成る。

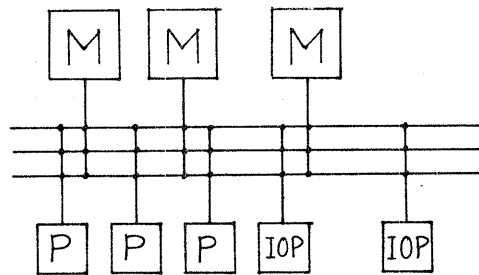


図4 並設多バス結合

結合方式 \ 拡張性項目	経済性	転送速度の拡張性	OSの容易性	障害に対する強さ
単一バス	◎	⊗	○	×
メモリ・スイッチ	×	△	○	△
ランダム結合	△	○	×	○
S-BUS	△	○	○	○

表1. 各結合方式の特徴

各々のモジュールの最大接続台数は、CMとIOMが合わせて32台まで、SMMは2台までである。S-BUSのバスの本数は原理上制限はないが、実装上4本までとしている。

コンピュータモジュール(CM)は演算処理を担当するモジュールで、'PULCE'と呼ぶマイクロプロセッサを用いている。⁽³⁾ PULCEとは7000ゲート以上を有するSOS(シリコンオンサファイヤ)のLSIで、内部16ビットのマイクロプロセッサである。PULCEは外部から32ビットのマイクロ命令を供給されてこれを解釈実行する。CMにはPULCEにマイクロ命令を供給するために、マイクロプログラムメモリ(MPM)、マイクロプロ

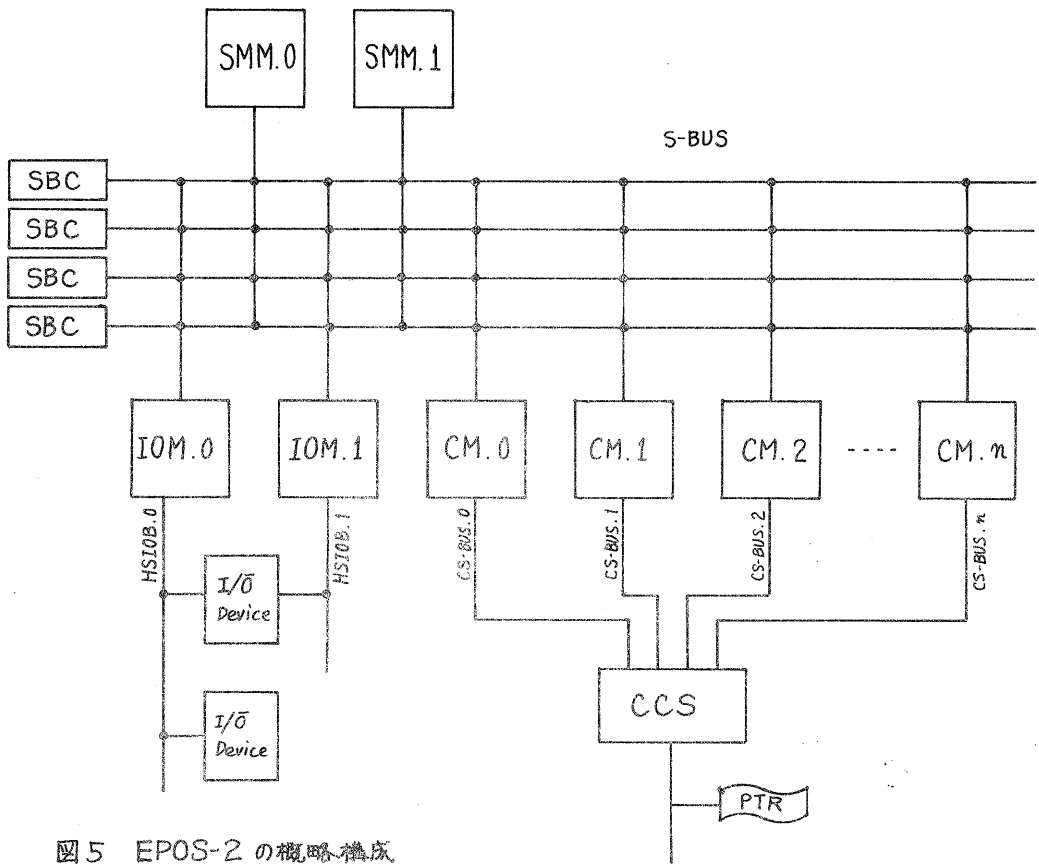


図5 EPOS-2の概略構成

グラムシーケンスコントローラ(MQC)がある。MPMIにはRAMとROMがあり、ダイナミックマイクロプログラムが可能で、CMを柔軟なものにしている。CMにはさらに、アドレスコントローラ(ADC)、ローカルメモリ(LM)がある。ADCにはメモリアドレスレジスタがありメモリへのアクセス要求を発生する機能のほか、2次元アドレスをサポートするための連想メモリとセグメントデスク립タレジスタ等がある。LMはCMに専用のメモリとして、SMMの競合の軽減を目的としている。ただし必要があれば他のCMのLMをもS-BUSを通して参照できる。EPOS-2のCMはEPOS-1のPMとマイクロプログラムレベルでできるだけ互換性を保たせて

ある。

IOMは、ハイスピードI/Oバス(HSIOB)の制御を行なうモジュールで、任意のメモリ内のチャンネルプログラムを内部のマイクロプログラムで解釈し、I/Oデバイスと任意のメモリとの間でデータの転送を行なう。このチャンネルプログラムはEPOS-1のIOCと互換性がある。

SMMは、CMやIOMの共用メモリとして、各種管理テーブル、共用データ、通信用バッファ等の格納に使用する。原則として各CMは可能なかぎり自身のLMを利用することによりSMMの負担を減らし2台に制限する設計思想を採っている。2台のSMMに対するアドレス分担には、単純な分割とインターリーブとが考えられるが、

インターリーブではかえってCM間の干渉と競合が増え、かつ一台の障害でSMMが全体として使えなくなるという欠点があるので、単純な分割を採用している。なおあるCMから見た場合、2台のSMMと自身のLMとは連続的にアドレス付けされている。

その他図5中のCCSは、ポリプロセッサシステムの電源の投入遮断、イニシャライズ、構成のモニター、接続切離しなどの際に、オペレータの負担を軽減するための集中コンソールである。CCSが障害を起した場合は、CCSがなくともポリプロセッサシステムは動作を続けることができるようになっている。

このようなEPOS-2に於て、S-BUSの役割は大きく分けて、メモリアクセスとプロセッサ間割込であり、それぞれ次のように分類できる。

■ プロセッサ間割込

- { CMから他CMかIOMへの割込
- { IOMからCMへの割込

■ メモリアクセス

- { 他CMのLMへのアクセス
- { SMMへのアクセス { SMM.0へのアクセス
- { SMM.1へのアクセス

S-BUSの各バスはSBCにより制御されるが、SBCは互いに独立しており、間を連絡する信号線などは無い。

[4] S-BUSの制御方式

すでに述べたとおり、CMを増設して処理能力を向上させるには、S-BUSの転送能力も増加の必要がある。そのためS-BUSではバスの本数を任意に増減できる必要がある。またプログラマーから見れば、このバスの本数は透明でなければならぬ。これらの要請を満たすためには、各バスは互いに独立している必要がある。

ここで各バスが互いに独立であるとは、それらが互いに連絡し合うこともないし、ある一つのバスが主導権を持つこともないし、さらに何かある共通の制御装置の下で制御されることもないということである。

各バスは互いに独立なSBCによって制御され

ているので、S-BUSを使用しようとするモジュールは各バス毎に要求を出す必要がある。

以後、S-BUSを使用して他のモジュールをアクセスないし割込もうとするモジュールを要求者、また、それに応答しサービスする相手側のモジュールを応答者と呼ぶこととする。

要求者がS-BUSを使用する一連の動作は大きく分けて、バス使用要求、使用バス決定、通信、の各動作に分かれる。図6はその一連の動作を流れ図で示したものである。

バス使用要求動作

バスの使用要求動作の概念を、バスが1本の場合について図7に示す。

SBCは複数の要求者に時分割で使用許可を与える機能が必要であるが、その分配は各要求者に対してできるだけ公平である方がよい。S-BUSではそのため、要求を一旦サンプリングする方式を用いている。

各要求者は各バス毎にバス使用要求信号を出

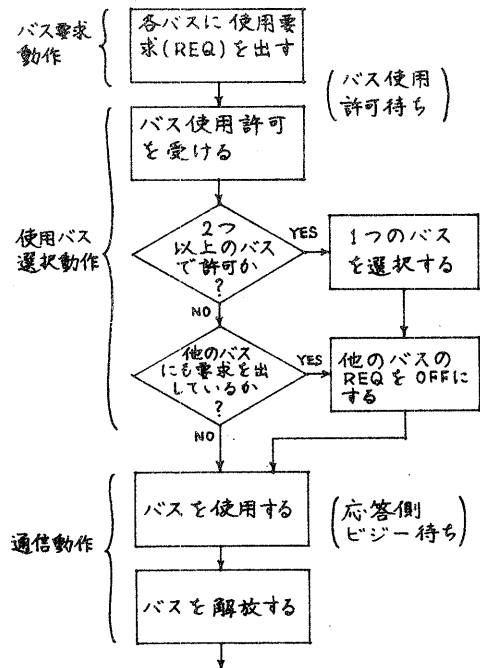


図6. バス要求者の動作

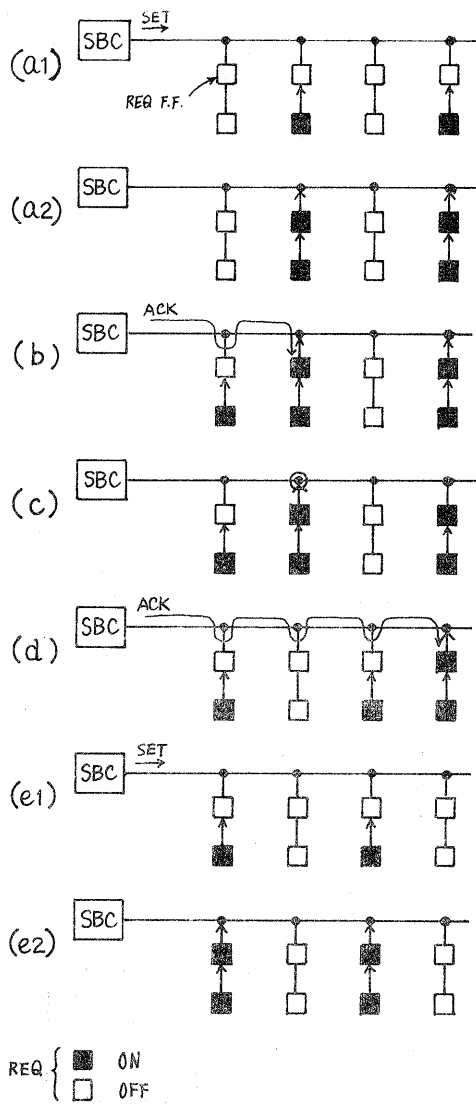


図7 バス要求受付方式

- (a1) バス上にREQ信号が出ていないと、SBCは要求をサンプリングするためSET信号を送る。
- (a2) SET信号が出ると要求者(モジュール2と4)はREQ F.F.をONとする。このF.F.の出力はWired-OrされてREQ信号となってSBCに与えられている。
- (b) SBCはREQ信号がONとなるとACK信号を出す。ACK信号は各モジュールのインターフェースをデイズーチェーンで結んでいる。この時点でモジュール1はバスを使用したいと思い始めたが、SET信号が終了した後のためREQ F.F.はOFFのままであり、ACK信号を通過させる。
- (c) モジュール2はREQ F.F.がONであるのでACK信号を受け取り、バスの使用権を得たことを知り、バスを利用して通信動作を行なう。動作終了後はREQ F.F.をOFFとする。
- (d) SBCはバス上での要求者と応答者の通信動作を見ており、終了時点を知ることができる。一回の通信動作終了後、まだREQ信号がONであると、SBCは再びACK信号を出す。モジュール3はこの時点でバスを使用したいと思い始めたが、SET信号が来ないのでREQ F.F.はOFFのままであり、ACK信号を通過させる。今度はモジュール4がACK信号を受け取り、バスを使用して通信動作を行ない、終了後REQ F.F.をOFFとする。
- (e1) REQ F.F.は全てOFFであるのでREQ信号はOFFとなる。SBCはこれを見て再び要求をサンプリングするため、SET信号を出す。
- (e2) SET信号により今度はモジュール1と3のREQ F.F.がONとなる。以下(b)~(e2)をくり返す。

すためのフリップフロップ(REQ)を有している。このREQがONとなるのはそのモジュールがS-BUSを使用しようとしていて、かつSBCからSET信号が来たときである。すなわちSBCはSET信号により要求者の要求をREQにサンプリングする。ONとなったREQからはバスへ使用要求信号がWired-orされて出される。SBCはこの要求信号を見て、使用許可信号を出して、

サンプリングされた全ての要求者にバスを使用させた後、再び次のサンプリングを行なう、という動作をくり返す。

一控サンプルされた要求はデイズーチェーン方式によりSBCに近い方から受けつけられて行くが、一旦サンプリングされた要求者の数は途中で増えることはないから、どの位置の要求者にも均等にバスの使用許可がなされる。

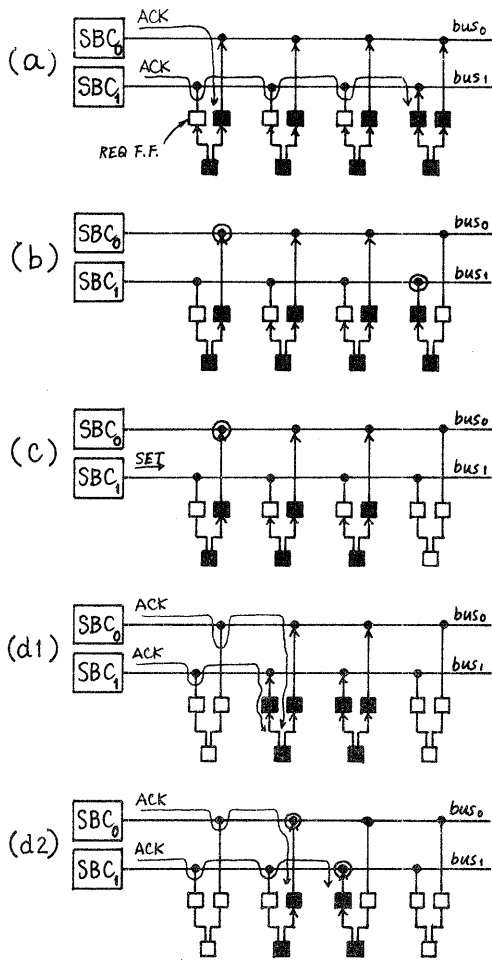


図8 使用バス決定機構

使用バス決定動作

要求者は複数のバスに使用要求信号を出す、それらのバスは同期しているわけではないので、その要求者への使用許可信号(ACK)はバス毎に別の時刻に来ることが多い。要求者は一番最初にACK信号の来たバスを使用することとし、他のバスへのREQ信号をOFFとする

しかし稀には同時に2つのバスからACK信号が到着することもあり得る。この場合には要求者は使用するバスを選択決定する必要がある。

図8にこれらの場合の決定動作を示した。各要求者はバス毎にREQフリップフロップを有している。REQを出しているバスからACK信号が

(a) SBC₀とSBC₁はそれぞれのバスのREQ信号がONなのでACK信号を出す。bus₀のACK信号はモジュール1が受け取る。bus₁のACK信号はモジュール4が受け取る。モジュール4はbus₀へもREQ信号を出しているが、bus₁が先に使用可能になったので、bus₀側のREQ F.F.をOFFとする。

(b) モジュール1はbus₀を、モジュール4はbus₁を用いて通信動作を始める。

(c) モジュール4は通信動作を終了しbus₁のREQもOFFとする。SBC₁はREQ信号がOFFであるのでSET信号を出す。モジュール1はまだ通信動作中であるが、すでにbus₀を確保しているのでbus₁のSET信号が来ても、bus₁側のREQ F.F.をONとしない。

(d1) モジュール1は通信動作を終了してREQ F.F.をOFFとする。SBC₀はまだREQ信号がONであるので次のACK信号を出す。SBC₁はREQ信号がONとなったのを見て、偶然同時にACK信号を出す。モジュール2へはbus₀とbus₁から同時にACK信号を受け取り、選択回路によりbus₀の方を使用することに決め、bus₀のREQ F.F.をOFFとし、bus₀のACK信号を次段へ送る。

(d2) モジュール3はbus₁からACK信号を受け取り、bus₁を使用することに決め、bus₀側のREQ F.F.をOFFとする。モジュール2はbus₀を、モジュール3はbus₁を用いて通信動作を開始する。

来たら、そのバスを使用することに決め、他のバスへREQを出していればこれをOFFとする。

同時にACK信号が来た場合には、それらの1つを選択し、他方のREQをOFFとして、そのACK信号を右側へ回送する。

このようにすると、あるACK信号はどのモジュールによって使用されず、かわりにREQ信号がOFFとなる場合があり得る。SBCはこのような場合にも正常に終了したものと認定する。

通信動作

通信動作は後に[5]に於て詳述する。

バスの使用モードの固定化

異なる要求者が異なるバスを通じて同時に同じ応答者に要求を出すと、そのうち一方はバスを確保したまま応答者が空くのを待たねばならない。この時さらに別の要求者が別の応答者への要求を持っていてもS-BUSを使用できないので待たされる。このように並設多バス方式では全てのバスが全ての応答者への通信に使用可能としておくと、かえってバスの使用効率及びCMの稼働率を低下させることがある。

今、S-BUSが3本のバスからなり、応答者が3個ある場合のCMの稼働率を簡単なモデルを使用して計算すると、図9のようになる。ここでAの場合は各バスとも全ての応答者への要求に使える場合、Bは3本のバスをそれぞれ3個の応答者への要求にしか使用しないと制限した場合である。図から明らかとなお、CMの台数が増えて競合が問題となるあたりからは、Bの方がAよりも30%ほど稼働率が高くなる。

このように、複数のバスがある場合は、それらの使用を応答者ごとに割り当てて制限する方がよいことが分かるので、S-BUSにもこの機能を導入した。

S-BUSでは使用モードとして、SMM.0へのアクセス、SMM.1へのアクセス、及びその他の3種に分けた。そしてこれらに対応した3本のア

ベイラブル信号を各バスに設け、SBCから直線的に出されている。

$$AF0 \begin{cases} = 1 & \dots \text{SMM.0のアクセスに用いてよい。} \\ = 0 & \dots \text{SMM.0のアクセスに用いてはいけない。} \end{cases}$$

$$AF1 \begin{cases} = 1 & \dots \text{SMM.1のアクセスに用いてよい。} \\ = 0 & \dots \text{SMM.1のアクセスに用いてはいけない。} \end{cases}$$

$$AFT \begin{cases} = 1 & \dots \text{他の要求に用いてよい。} \\ = 0 & \dots \text{他の要求に用いてはいけない。} \end{cases}$$

例えばあるバスのアベイラブル信号が、 $AF0=1$ $AF1=0$ 、 $AFT=0$ であれば、そのバスはSMM.0へのアクセス要求のあるモジュールだけがREQ信号を出せる。

このような機構を実現するために、図10に示すごとく、各要求者のREQフリップアフロップのセット条件として、自分のその時の要求のタイプとそのバスのアベイラブル信号との一致が必要ようになっていく。

SBCからのアベイラブル信号は、手動、またはCCSからの指令で任意に変えられるようになっている。3本以上のバスが動作可能であれば、通常そのうち1本をSMM.0専用、他の1本をSMM.1専用として使用する。もし1本のバスしか動作可能でなければ、そのバスのアベイラブル信号は全て1とし、全ての使用モードを許可する。

このようにS-BUSでは各バスの使用モードを

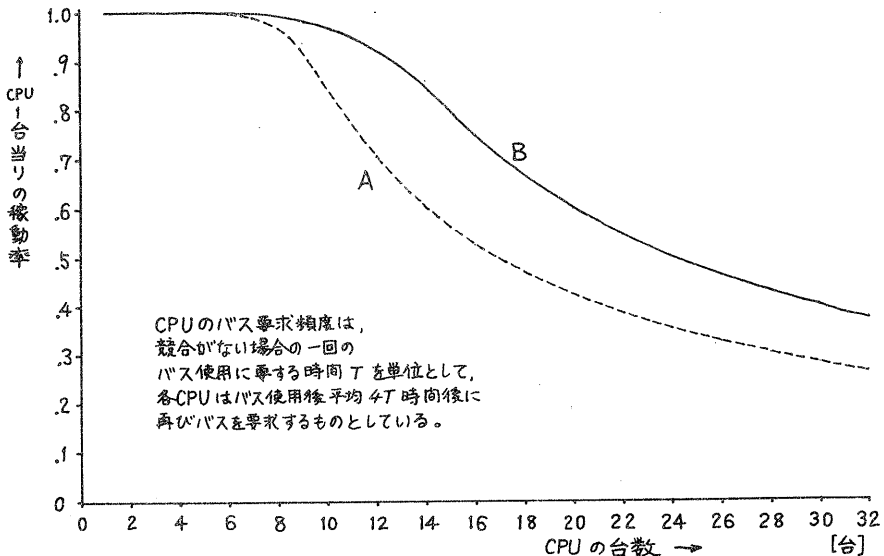


図9 バスの使用モード固定の効果

制限し得るようにしたことにより，拡張性とバスの効率的な使用とを両立させている。

[5] 各バスの通信動作

各バスの信号線

S-BUSの各バスはおよそ50本の信号線から成り，その内訳は表2に示すごとくである。

データの転送は，32本の信号線(SDAT)を通して32ビット並列で行なわれる。この信号線には1ビットのパリティ(SBPR)が伴っている。

一方メモリアクセス時のアドレス線としては，

- ① これを別に設けると信号線数が多くなりすぎ，実装上バスの設置本数が制限される。
- ② メモリアクセスの90%は読出し動作であり，読出し動作ではアドレスとデータを時分割して送受してもそれほど遅くならない。

という2つの理由で32本の信号線(SDAT)とアドレス線を共用することとした。

また，IOMが高速の周辺機器，例えばディスク記憶装置とSMM等とデータ転送を行なっている場合，1回のSMMのアクセス毎にS-BUSを確保すると転送が遅くなってデータが脱落するおそれがある。この場合にはSCON信号を用いてS-BUSのうちの1本のバスを連続使用できるようになっている。

通信動作とタイミング

バスの使用効率の高い通信方式としては，ア

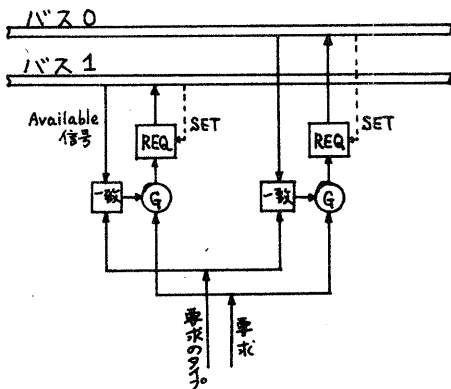


図10 使用モード固定機構

ドレスの送信とデータの送信との間で一旦バスの使用権を解放する方式や，さらに応答側がビジーであっても要求を送ってバッファで受ける方式などが考えられる。しかしこれらの方式では，

- ① 受け側にバッファが必要で，送り側，受け側ともに回路が複雑になる，
- ② バッファが先着要求で満席である場合にはやはり要求の送信を待たせねばならない，
- ③ 通信路を一旦切るとその後要求内容にエラーが発見された場合の取り扱いがむずかしい，

等の理由で，S-BUSでは一回の要求動作は完了するまでバスを確保しつづける方式とした。

さてREQ信号を出したバスからACK信号が来ると，要求者はそのバスを確保して通信動作を開始する。

通信動作には，1サイクル型と2サイクル型の2つの動作の形がある。

- { 1サイクル型…… 割込要求
- { 2サイクル型…… メモリアクセス要求

データ送受方式は非同期型であり，

○1サイクルはASYNとMSYNにより，

○2サイクルはSSYNとESYNにより，

タイミングを合わせせる方式となっている。

1サイクル型でも2サイクル型でもその最初のサイクルでは次の要求情報をSDAT上に乗せる。

- { 要求のタイプ
- { 相手のモジュール番号
- { メモリアドレス
- { アクセスの単位(全語/半語)

これらの内容は要求のタイプ毎に図11で示されるフォーマットでSDAT線に出される。

ここで6ビットのモジュール番号は各CMとIOMにつけられていて，他CMのLMをアクセスする場合と，プロセッサ間割込を行なう場合に相手側(応答者)を指定するのに使用する。

メモリのアクセスの単位は全語(32ビット)と半語(16ビット)があって，その区別をLSD(F)で行なっている。

一方各モジュールはバス上に他のモジュールが出した要求情報を常に監視しており，自分が

応答者であると認識すると、ただちに MSYN 信号を ON とする。もし自分がビジーである場合は MSYN 信号を ON としつづける。その後この要求を受けつけられる状態になったら MSYN を OFF とする。要求者は MSYN が ON となると応答者が存在していると認識し、OFF となると自分が受け付けられたことを知る。図 12 にこれら一連の通信動作のタイムチャートを示した。

同図中信号は Negative True で表記している。
メモリアクセスの場合には次に書込データまたは読出データの送受サイクルが入る。

エラー対策

S-BUS 上にはいろいろな誤動作の発生が想定できるが、S-BUS では次のような異常検出機能が用意されている。

信号略号	方向	結線	意味/動作	
AFO	SBC → 要求者	単方向	Available for SMM 0	SMM 0 のアクセスにこのバスを用いてよい
AF1	SBC → 要求者	〃	Available for SMM 1	SMM 1 〃
AFT	SBC → 要求者	〃	Available for Others	LM のアクセスと割込に 〃
SET	SBC → 要求者	〃	Set	REQ のサンプル信号
REQ	SBC ← 要求者	W.O.	Request	バス使用要求
ACK	SBC → 要求者	単方向デジタル	Acknowledge	バス使用許可
APAS	SBC ← 要求者	W.O.	Acknowledge Passing	自分は ACK 信号を次段に送った
ASYN	要求者 → 応答者	Tri.S.	Address Synchronous	SDAT 上に要求情報 (Address) を出した
MSYN	要求者 ← 応答者	Tri.S.	Match Synchronous	応答すべきモジュールがあった
SSYN	要求者 → 応答者	Tri.S.	Start Synchronous	(タイムチャート参照)
ESYN	要求者 ← 応答者	Tri.S.	End Synchronous	(タイムチャート参照)
SDAT	要求者 ↔ 応答者	Tri.S.	Address/Data	要求情報 / Read, Write Data
SDPR	要求者 ↔ 応答者	Tri.S.	Parity	パリティ
SME0	要求者 ← 応答者	Tri.S.	Memory Error	アドレス不当
SMPRT	要求者 ← 応答者	Tri.S.	Memory Parity Error	メモリ側パリティエラー
SIFO	要求者 → 応答者	Tri.S.	Interface register 0	半語転送での上位/下位指定
STO	SBC → 要求者	単方向	Time Out	タイムアウト発生
SPRT	SBC → 要求者	単方向	Parity Error	パリティエラー検出
SCON	要求者 → 他要求者	W.O.	Continuous	S-バスの連続使用宣言

表 2. バス 1 本当りの信号線

(SDAT ビット番号)		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
他CMへの割込	CMから	0	0	0	MNO.																												
	IOMから	0	0	1	MNO.																												
他CMのLM	Read	1	0	0	MNO.														Address										F				
	Write	1	0	1	MNO.														Address										F				
SMM.0	Read	0	1	0	MNO.														Address										F				
	Write	0	1	1	MNO.														Address										F				
SMM.1	Read	1	1	0	MNO.														Address										F				
	Write	1	1	1	MNO.														Address										F				

(OPコード)

MNO. --- モジュール番号

F --- Half word Access か Full word Access かの区別

図 11 要求情報のフォーマット

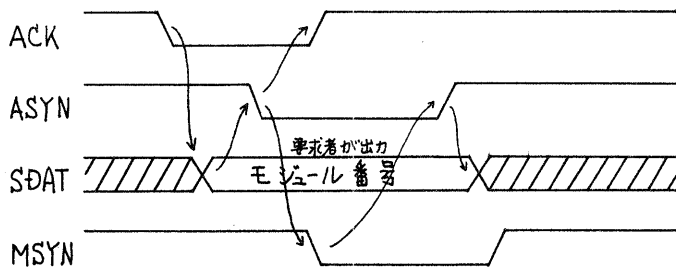


図12 (a) 割込動作のタイムチャート

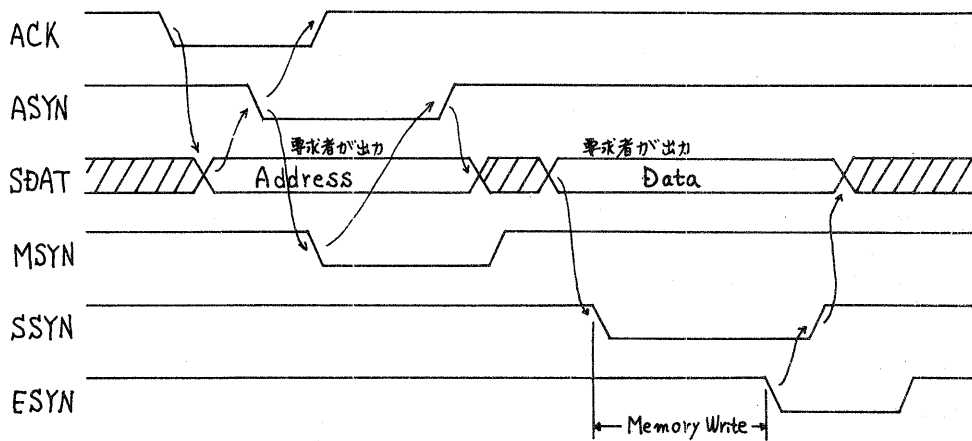


図12 (b) Memory Write 動作のタイムチャート

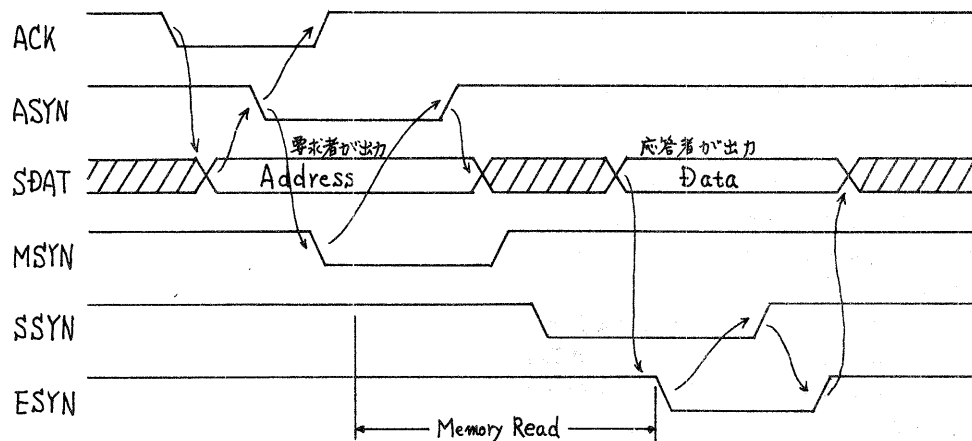


図12 (c) Memory Read 動作のタイムチャート

タイムアウトの検出----SBCはバス上の要求情報から1サイクル型か2サイクル型かを認識した上でASYN, MSYN, SSYN, ESYNの各信号を監視し、一定の時間内に通信が終了しないとタイムアウト機能が働いてSTO信号をONとする。要求者と応答者はSTO信号がONとなったら以後の動作は中断する。

アドレス異常の検出----各メモリ(LM, SMM)は自己の割付範囲外のアドレスを受け取るとSMEO信号により要求側に知らせる。

メモリパリティエラーの検出----各メモリは内部でパリティチェックを行っており、読出時にエラーが検出されるとSMPRT信号により要求者へ知らせる。

バス上のパリティ検査----SDAT上の要求情報とデータには常にパリティをSDPRにつけることになっている。その検査はSBCでも要求者、応答者でも行なう。SBCがエラーを検出した場合にはSPRT信号をONとする。SPRT信号がONとなったら要求者と応答者は通信を中断する。

これらの異常検出結果は全て要求者に対する割込原因となるようになっている。

[参考文献]

- (1) 田丸 他 「実験用ポリプロセッサシステム, EPOS-1 について」 信学会 EC75-67
- (2) P.H. Enslow Jr. 'Multiprocessor Organization - A Survey' Computing Surveys. Vol.9, No.1, 1977
- (3) 「パターン情報処理システムプロジェクト・マイクロプロセサ PULCE 解説書」 電子技術総合研究所 昭和51年1月
- (4) 田中 他 「ポリプロセサ・オペレーティングシステム: EPOS の構造」 情報処理学会第17回全国大会 No. 169. (1976年11月)
- (5) 森本 他 「APL インタプリタにおける演算処理」 同上 No. 280

[6] むすび

拡張性を重視したポリプロセッサシステムの結合法として、EPOS-2 で採用したS-BUSについて、その設計思想と内部制御方式について述べた。S-BUSは一種の独立並設多バス方式とも言える方式である。使用バスの選択を要求側で行なうことにより、バス間の独立性を実現し、その結果バス本数の増減が容易となった。また各バスの使用目的を応答者別に割り当てることを可能とすることにより、バスの利用率を最適に保つことを可能とした。

本ポリプロセッサ・システム上で動くソフトウェアはハードウェアと並行して開発が進められており、OS, FORTRAN, APL, PASCAL などを準備中である。(4), (5)。

なお、本ポリプロセッサ・システムは通産省工業技術院の大型プロジェクト「パターン情報処理システムの研究開発」の一環として開発中であり、最終的には“リングバス”により他の認識システム等と結合されて“パターン情報処理総合システム”を構成する予定である。