

非同期制御回路網 (ACN)

ASYNCHRONOUS CONTROL NETWORKS (ACN)

並列制御用ハードウェアの記述

Description tool for parallel control hardware

増 山 博

Hiroshi MASUYAMA

(広 島 大 学 ・ 工 学 部)

Faculty of Engineering, Hiroshima University

1. まえがき

システム内で、互いに独立に、又は従属的に処理すべき複数のプロセスを、クロック信号による同期化しに制御する方法について多方面から検討が行われている⁽¹⁻⁶⁾。これらの研究は、大別すれば2つの方向、(1)非同期処理プログラムなどでのPV操作⁽⁹⁾に代表されるソフトウェアによる処理⁽¹⁻²⁾、(2)ハードウェアによる処理⁽³⁻⁶⁾に分けられる。後者のものをさらに制御方式から2分すれば、幾つかの機能モジュールを流れ図に従って接続してモジュール単位でプロセスを制御する方法⁽¹⁰⁻¹¹⁾と、状態遷移表に従った状態遷移で生ずる状態を単位としてプロセスを制御する方法とに分けられる。マイクロプログラム化制御回路網の原点とみられるHudson⁽⁴⁾の方法は状態を単位とした制御の方法に属するし、非同期順序回路⁽⁵⁾による方法はこの1例である。非同期順序回路論に従った、いわゆるセミモジュール⁽⁷⁾なシステムの設計は数学理論的には興味深い⁽¹²⁻¹³⁾が、並列制御を状態遷移表で表現することは一般的に難しく、かつ現実的には複雑な並列制御にばるほど入力状態の設定が複雑となる。このため並列制御の場合には、一部の簡単な制御の場合を除いて、機能モジュールを相互接続して構成されるシステムが通している。ところで通常の入力とは別に自己の出力をリセットできるリセット入力をもつモジュールを用いると複雑な並列制御が表現できる。このことは、このモジュールの数学的モデルと見出すことのできるペトリネット⁽⁸⁾及びマークグラフ⁽¹⁵⁾

等の有能性より明らかであり、順序回路で実現可能なデジタルグラフは単純であることから判定できよう⁽¹⁶⁾。このように適用能力の高いいわゆるモジュラー非同期回路網には、ハードウェア使用量が大きい難点がある。従って、出来るだけ整理された少ない種類のモジュールの相互結合回路網で、しかも適用能力が低下しないことが望ましい。

この論文では、2種の機能モジュールを相互接続して構成されるシステムによりプロセスを非同期的に制御するためのハードウェアの記述とその実現上の問題について議論し、2,3のシステムへの適用について検討する。すなわち、2章ではプロセスを速度独立に制御するプロセス制御モジュールを与えて、従来周知のマージモジュールとの相互接続によって構成される非同期制御回路網(Asynchronous Control Network; ACN)を定義する。3章では、ACNの構造と動作について議論する。一般に、各モジュールのリセット入力により各出力端子の信号を独自にリセットする回路網ではハザード問題⁽¹⁷⁾をそれほど考慮する必要はないが、反面デッドロック⁽¹⁸⁾(又はブロッキング⁽¹⁹⁾)現象が発生する危険がある。ここでは、これを避けるための構成条件について議論し、プロセス制御モジュールのリセット方式により条件が満たされることを示す。このため、システムの設計は容易である。4章では、単に並列命令及び分岐命令が実現できるだけでなく相互排除命令をも実現できることを示すため、通常のアービタとこれに基づく多質素アービタを構成する。さらに、具体

的なACNの適用システムとしてマルチプロセッサシステム用の制御システムについて検討する。

2. モデルによる記述

本論文で使用する用語「プロセス」, 「モジュール」及びプロセスを制御する「ACN」を定義する。さらに, ACNを並列制御用ハードウェアとしてのユーザーレベルの記述に適したモデルとするため, フローチャート形式で表現できるようにする。以下において, 特に断らば限り使用する変数は2値であり, 式はブール式である。

〔定義2・1〕 プロセス P_i とは, 次の性質をもつ順序対 $P_i = (b_i, e_i)$ である。即ち, P_i は e_i が b_i に依存した (b_i, e_i) を状態とした以下に示される2つの状態遷移のいずれかによってその動作が表現される; (1) 静止状態 $S_1(0, 0)$ の後, 要求状態 $S_2(1, 0)$ から応答状態 $S_3(1, 1)$, さらにリセット状態 $S_4(0, 1)$ へ遷移した後 S_1 へ戻る(図1の実線を示される), 又は(2) S_1 から直接 S_3 へ遷移した後直ちに S_1 へ戻る(同図の点線を示される。)(2)の場合のプロセスを特に空プロセスと称する。

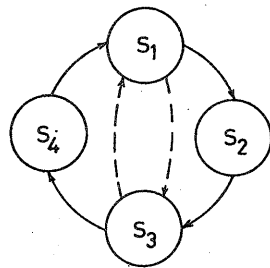


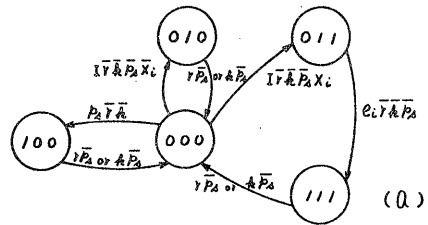
図1. プロセスの持つ2種類の遷移

本稿では, 以下変数とこれを示すモジュールの端子を同一記号で表わす。

〔定義2・2〕 プロセス制御モジュール P_i とは, 外部信号として入力 I , プロセス制御条件 X_i ($i=1, 2, \dots, n$), プロセス P_i の第2変数 e_1, e_2, \dots, e_n , リセット r , クリア a 及びリセット b を受理する端子 b_i , P_i からの発生信号として P_i の第1変数 b_1, b_2, \dots, b_n 及び出力 O を発生する端子をもち, 次の機能をもつ。 $I=1$ により入力受理変数 a を1にセットし, $a=1$ かつ $X_i=1$ のとき b_i を1にセットする。次に, $a=1$ かつ n 個の中のいずれかのプロセスの第2変数が1のとき O を1にセットする。また, r 又は $b=1$ のとき O を0にリセットでき, I 及び n 個のプロセスとは独立に

$P_i=1$ により O を1にセットできる。 X_i が1(固定値)なる i が存在する P_i を特に条件 P_i といひ P_{rc} と記す。又 $n=1$ かつ常に $b_i=e_i$ なる P_i を $P_i(\Phi)$ とし, 条件 $P_i(\Phi)$ を $P_{rc}(\Phi)$ と記す。

P_i における条件 X_i とは, 条件変数の集合 $\{x_1, x_2, \dots, x_n\}$ の要素による論理関数で与えられる。 \dagger 又 n の値は必要に応じて決定される。 P_i について (O, b_i) を状態として表わした状態遷移図で与えられる動作モード, セル及び P_{rc} の記述形式をそれぞれ図2(a), (b)及び(c)に示す。変数 a を表わす端子は必要に応じて記述できるとする。



状態 = (O, b_i)

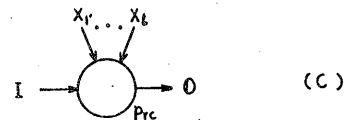
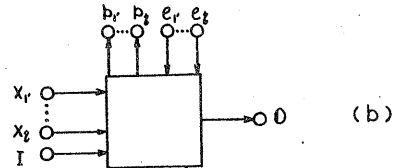


図2. プロセス制御モジュールの(a)動作モード, (b)セル, (c)記号

〔定義2・3〕 マージモジュール M とは, シリアル M (SM)とパラレル M (PM)とに分けられ, 変数 y_j ($j=1, \dots, m$)に対してその出力 Y を次のように与えるモジュールである。

$$SM : Y = y_1 + y_2 + \dots + y_m$$

$$PM : Y = y_1 \cdot y_2 \cdot \dots \cdot y_m$$

ここで, y_j 及び Y は P_i における変数 I, O, X_i, a, b_i 及び r のいずれかを取ることができ。

† ACNにおける条件 X_i を具体的に機構化するためには通常否定素子をモジュールとして加えればよいことは明らかであろう。

$m=1$ なるMは、単純Mと呼び単に結線を意味することにする。

〔定義2・4〕 ACNとは、 P_i 及びMからなる相互結合回路網である。

上で明らかのように、1つの P_i は数個のプロセスの処理の開始信号を出し、終了信号を受け取るモジュールとみることができる。ACNはプロセスの集合 $\{P_1, P_2, \dots, P_n\}$ を制御するモジュール結合回路網であり、一般構成図は図3で表わされる。ここで、各プロセスを1つの遅延とみればACNは速度独立な回路として取扱うことができる。

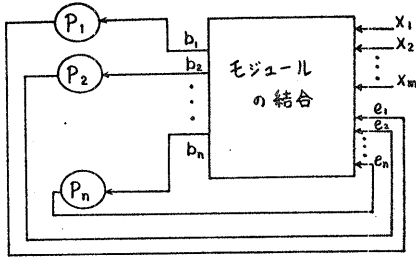


図3. ACNの一般構成図

ACNの実現する制御モードの代表的な例として、並列命令 \mathbb{I}_p 、分岐命令 \mathbb{I}_b 及び相互排除命令 \mathbb{I}_e の記述形式を図4に示す。

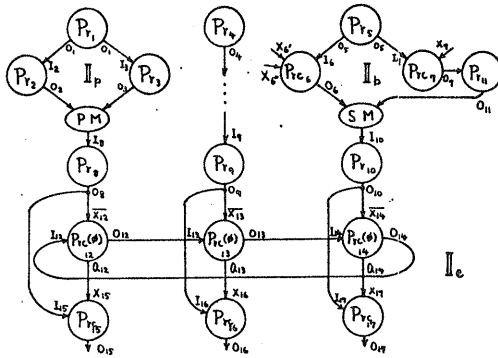


図4. 並列命令 \mathbb{I}_p 、分岐命令 \mathbb{I}_b 及び相互排除命令 \mathbb{I}_e のACNにおける記述例

次章では、ACNの動作モードと、これによって忠実にシミュレーションできる他のモデルとの対応関係などについて議論する。

3. 構造と動作モード

非同期回路の1つの類別法として、出力信号のリセット方法による分類が考えられる。非同期順序回路についても機能モジュールによる構

成が考えられるようになって来たが⁽⁹⁾、これを含めて順序回路方式は前状態と入力によってリセット入力が設定される意味から同様に属するとみることができ、実現の問題は2値モジュール回路を構成する問題に帰着される⁽⁴²⁾。モジュラー型回路網の場合には、誤動作を回避するための色々のリセット入力の印加のタイミングが考えられる。本章では実用的なリセット方法を考察し、かつこれを満たすモジュールを具体的に提示する。さらに、具体化されたACNと他の既知のモデルとの関係について言及する。

ACNにおいて、1つのMの入力端子と直接接続された P_i (P_{iM} とする)を、このMの出力端子と直接接続された P_j (P_{jM})の先行者といい、 P_{jM} を P_{iM} の後続者という。ACNにおける P_i の出力変化の因果関係を示すため、通例の有向グラフを用いる。 P_i を節点とし、 P_i の接続関係を有向枝で示した有向グラフをGとする。ACNにおける1つの P_i の出力変化 $0 \rightarrow 1$ は、この P_i の後続者の出力がすべて0、かつこの P_i と先行者間がPMのとき先行者の出力がすべて1であるとき許可され、この P_i と先行者間がSMのとき先行者の1つの出力が1のとき許可されるとする。1つの P_i の出力変化 $1 \rightarrow 0$ は、この P_i の後続者がすべて条件 P_r 又は条件 $P_r(\phi)$ のときこの中の1つの出力が1のとき許可され、そうでないとき、後続者のすべての出力が1のとき許可されるとする。これ以外の組合せはACNでは存在しないとする。節点dに対応する P_i にPMが接続され、後続者は条件 P_r 及び条件 $P_r(\phi)$ でない場合を図5に例示する。ここで、ACNの設計の出发点を正当性のある⁽⁴⁷⁾フローチャートと考え、文献(48)ほどのようなユーザレベルのデッドロックは存在しないとする。

上述した出力変化の条件の下では、図6(a)に例示するGにおいて、同図(b)のデジタルグラフの示すように各節点の出力がグラフのパスに依存した異なった値を履歴しく非同期理論では、確定的⁽⁴⁹⁾でないという)、ACNとしてはプロセス制御に誤りをもたらし、このような誤動作(モジュラー型回路網というデッドロ

ト) 相互排除命令は記述形式から判別できるようにリング式アービタと同一の機能構造となっている。

ック又はブロッキング)は、次の構成条件Aにより回避される。[†]

〔構成条件A〕 すべて P_r について、 P_r の出力変化 $0 \rightarrow 1$ からリセットによる $1 \rightarrow 0$ までにかかる時間が、 P_r の入力印加 (I の $0 \rightarrow 1$ の遷移)からこれに起因した自己の出力変化 $0 \rightarrow 1$ までにかかる時間より大きくない。

構成条件Aの下では、図6(b)の*印のパスは存在せず、確定的である。また、一般的なACNにおいては、 P_r の1出力は再び同一の P_r に入力を印加するまでにリセットすることが必要で、このことはACNの速度独立性を保つ上で重要⁽¹⁾かつ常識的⁽²⁾なことであるが構成要件Aによって保証される。

上の議論より明らかに、 P_r のリセット入力端子には、この P_r のG上の後続者が条件 P_r 又は P_r (中)のときSMをそうではないときPMを接続し、これらのMの入力は後続者の α 信号とすればよい。

〔定義3・1〕 G上に2節点のみ存在する相向グラフを単純ループと呼ぶ。

図7に P_r の具体的な回路図を2つ与える。こ

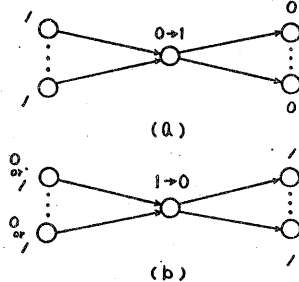
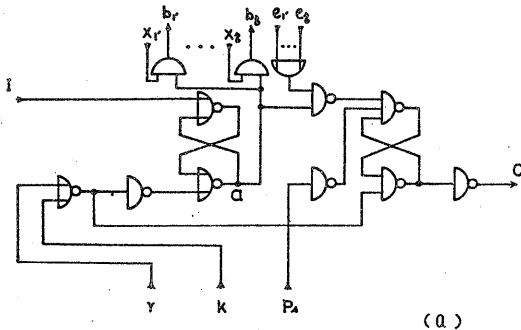


図5. プロセスモジュールに出力変化を転写する状態



(a)

これらの P_r は、単純ループで表わされるACN以外のACNにおいて、上述したリセット方式の下では構成条件Aを満たすことができる(P_r (中)を考慮すれば十分である)。図7(b)の P_r はプロセスからの終了信号がパルス信号ではなくレベル信号である時のみに有効であるが、きわめて簡潔である。

以下、ACNは図7の回路を用いるものとする。

モジュラー型回路網の中には、ACNのように直接ユーザレベルのフローチャートをモデルとして実現するタイプ; Stucki⁽¹⁰⁾, Nordman⁽¹¹⁾, Keller⁽¹²⁾等と、一旦数学的モデルであるマークグラフ又はペトリネットで表現した後実現するタイプ; Jump⁽¹⁵⁾, Dennis⁽¹³⁾, Patil⁽¹⁴⁾, Misunas⁽¹¹⁾等のものがある。NordmanのモデルはIlliac IIIシステムを対象としたもので、使用されるモジュールの基本機能はACNと同じとみてよい^{††}。Kellerの言う意味で、 P_r モジュ

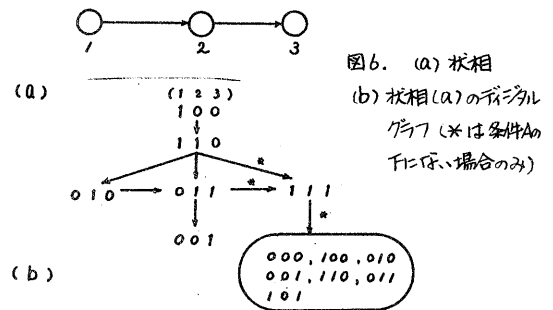
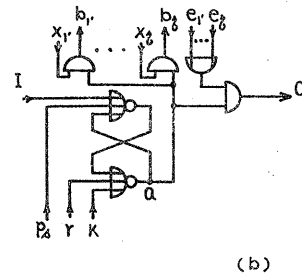


図6. (a) 状態

(b) 状態(a)の信号グラフ (*は条件Aにばい場合のみ)



(b)

図7. プロセス制御モジュールの2種類の基本論理回路(a)(b)

†) 出力変化 $1 \rightarrow 0$ の条件が先行者の出力に依存する場合⁽¹⁸⁾に比べて、構成条件が簡単である。

††) モジュール自体に故障診断機能が与えられている。

ールにより Select モジュール⁽²²⁾の機能が表現できるので、ACN はすべてのリアル・モジュールを実現可能であり、又 P_r と M 両モジュールにより Arbitrating Test-and-Set (ATS) モジュール⁽²²⁾の機能が表現できるので、ACN はすべてのパラレル・モジュールをも実現可能である^卅。Stucki のモデルは、FORK (BRANCH を含む) 及び JOIN のみの言語で記述されるマイクロプログラム⁽²⁵⁾等^卅の一般性のある制御機能のみを有しており、ACN で実現可能であり、両モデルともモジュールが簡単なのでレジスタ転送用モジュール⁽²⁴⁾としても直接利用できる。

〔定義3・2〕 ACN がセーフペトリネットを実現するとは、セーフペトリネットのトランジションと ACN のプロセス (空プロセスを除く) が 1対1 に対応し、かつセーフペトリネットのトランジションの発火順序と ACN の応答状態をとるプロセス (空プロセスを除く) の順序が同一であることを示す。但し、1対1 対応の下ではラベルは同一なものとする。卅

マークグラフ又はペトリネットの各要素を直接モジュールで構成してセーフマークグラフ又はセーフペトリネットを実現した場合に、相当性のある適用分野はごく限られたものであるが (例えば、Communication Protocol に近い分野^(31~32))、ACN ではペトリネットのプレースを 2 モジュール P_r (中) と SM 及び相互排除命令の記述構造を利用して構成し、トランジションは P_r と PM を利用して構成すれば、ペトリネットを実現できるので、記述の複雑さを無視すれば前出のモデルの能力を有している。また、マークグラフは Muller C ゲートと NOR ゲートによって実現可能であるが⁽³⁵⁾、ACN でも明らかに実現可能である。

4. 適用システム例

ハードウェアによる非同期プロセス制御の適用例としてマルチプロセスシステムを考える。このために利用するアービタをまず ACN により、次章で示される ACN による n 入力 1 資源アービタの n 個の各セル又は図4での P_{rc}(φ) は A_{ts} モジュールの機能を有する。
 卅) ACN はシリアル及びパラレルモジュールのクラスに対してユニバーサル⁽²²⁾である。

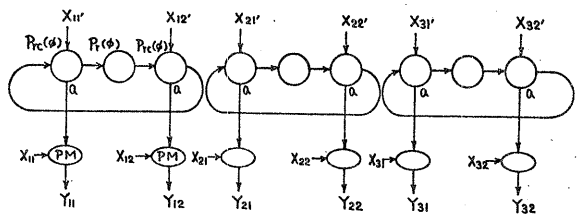
って具体的に実現する。

4・1 多入力 1 資源アービタ

3 入力 1 資源アービタの ACN による記述形式は図4での命令 I_e で与えられている。ただし、アービタとしては同図での Q₁₅ ~ Q₁₇ をプロセッサの資源要求 (Q₁ ~ Q₁₀) 信号に対する応答信号と考えればよいので、P_{rc15} ~ P_{rc17} は 2 入力の PM で置き換えてよい。n 入力 1 資源アービタの具体的な論理回路は文献⁽³¹⁾で与えられている。

4・2 多入力多資源アービタ

多資源アービタの 1 例として 3 入力 2 資源アービタを図8に記述する。同図において、R_i 及び A_{ij} はそれぞれ (例えばプロセッサ j) i の資源に対する要求信号及び i への資源 j の応答信号である。図中の 3 つのループはそれぞれが相互排除命令の記述形式を示している。3 入力 2 資源アービタの具体的な論理回路は付録に示す。なお、動作は確認されている。



$$\begin{aligned}
 X_{11} &= R_1 \bar{Y}_{12} \bar{Y}_{13} & X_{21} &= R_2 \bar{Y}_{11} \bar{Y}_{13} & X_{31} &= R_3 \bar{Y}_{11} \bar{Y}_{12} \\
 X_{12} &= R_1 \bar{Y}_{22} \bar{Y}_{23} & X_{22} &= R_2 \bar{Y}_{21} \bar{Y}_{23} & X_{32} &= R_3 \bar{Y}_{21} \bar{Y}_{22} \\
 X_{ij} &= \bar{X}_{ij} \\
 R_j &= \text{プロセッサ } j \text{ の資源に対する要求信号} \\
 Y_{ij} &= \text{資源 } i \text{ のプロセッサ } j \text{ に対する応答信号}
 \end{aligned}$$

図8. 3入力2資源アービタの ACN による記述

4・3 マルチプロセッサシステムの制御回路

マルチプロセッサシステムの記述は文献⁽³⁴⁾に詳しいが、最近のシステムに適した記述をすれば以下のようになる。ここでは紙面の都合上エントリポイント数4、メモリバンク数4に限定して、共有バス方式とスイッチマトリックス方式の2方式について記述する。

〔システムの記述〕

卅) グラフ理論というDチャート⁽³⁶⁾のプログラムが含まれる。
 卅) 通常行われる厳密な数学的「実現」の定義は付録で与えられる。

共有バス方式及びスイッチマトリックス方式
共に次の4要素により構成され、構成図はそれぞれ図9及び図10(データ部については省略しているが同様とする)に示す。

μ; メモリバンク

β; バス

γ; エントリポイント

δ; アービタ

α) メモリバンク(M_i) (i = 1, 2, 3, 4)

は次のレジスタを有する。

MA_i; アドレスレジスタ(16ビット)

MD_i; データレジスタ(16ビット)

ステータスレジスタ(2ビット; ビット名

MR_i, MW_i)

MR_i = 1 のとき, 読み出し可能を表示

MW_i = 1 のとき, 書き込み可能を表示

β) バスは次の種類を有する。

ADBUS; アドレスバス

DBUS; データバス

メモリバンク及びエントリポイントのレジスタ間の情報の伝達を制御するためにバスドライバーSA_iと, SD_iと(γ = 1, 2)がある。

γ) エントリポイント(Entry point i)は次のレジスタをもつ。

CA_i; アドレスレジスタ(16ビット)

CD_i; データレジスタ(16ビット)

ステータスレジスタ(2ビット; ビット名

CR_i, CW_i)

CR_i = 1 のとき, 読み出し操作を表示

CW_i = 1 のとき, 書き込み操作を表示

BS_i; バンクセレクション(2ビット)

メモリバンク番号を2進表示

R_i; 操作待ち命令レジスタ(1ビット)

R_i = 0 のとき, 待ち状態に設定される。

δ) アービタは次のレジスタをもつ。

R; 要求命令受理レジスタ(4ビット; ビット名r₁, r₂, r₃, r₄)

A; 応答命令レジスタ(4ビット; ビット名a₁, a₂, a₃, a₄)

上で記述されたシステムにおいて, プロセッサの読み出し制御のための流れ図を例示すると図11となる。ここで, メモリバンク2~4に関するところは省略されている。これを基に, マ

ルチプロセッサシステムのための制御をACNで記述すると共有バス方式が図12により, 又スイッチマトリックス制御方式が図13で与えられる。具体的論理回路は以前の議論より明らかなので省略する。

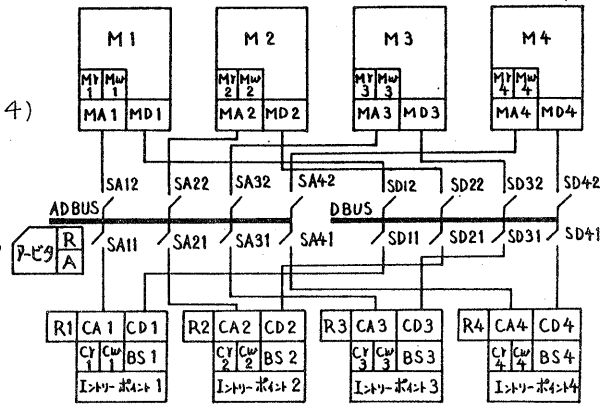


図9. メモリプロセッサ共有バス構成

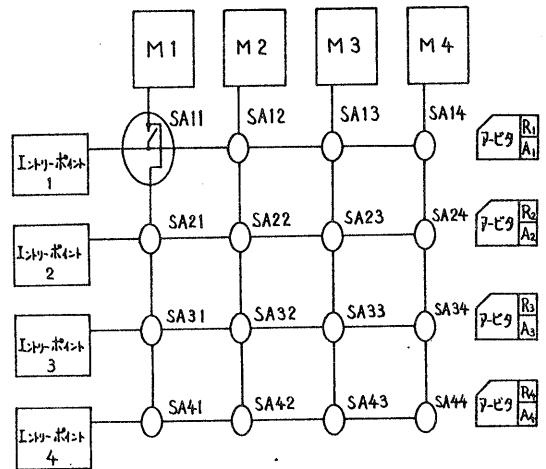
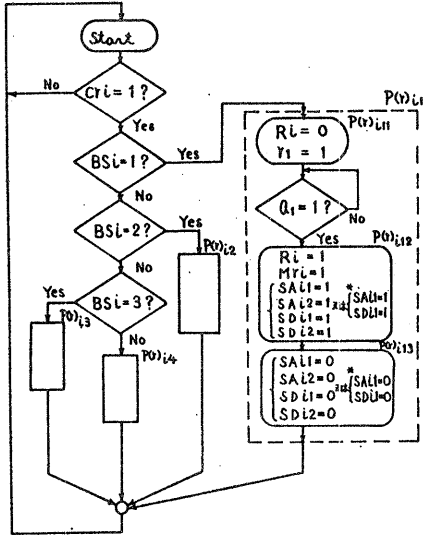


図10. メモリプロセッサスイッチマトリックス構成

5. まとめ

1つの演算である「プロセス」の開始命令を非同期に制御するためのハードウェアとして非同期制御回路網を定義し, これの適用システムとして具体的にマルチプロセッサ制御回路を例示した。又同方面の2, 3の既知のモデルとの関係についても言及した。

本稿の非同期制御回路網はモジュラー形式の



P(i)ij : プロセッサのメモリバンクに対する読み出し操作
 * : スイッチマトリクス構成の場合

図11. プロセッサの読み出し制御のための流れ図

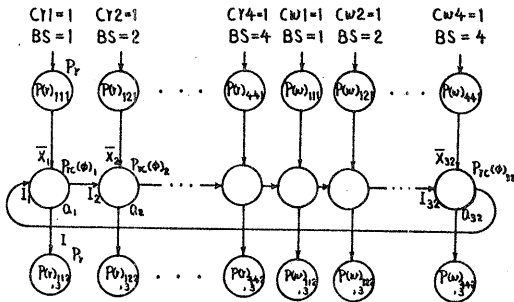


図12. 共有バス制御の命令構造

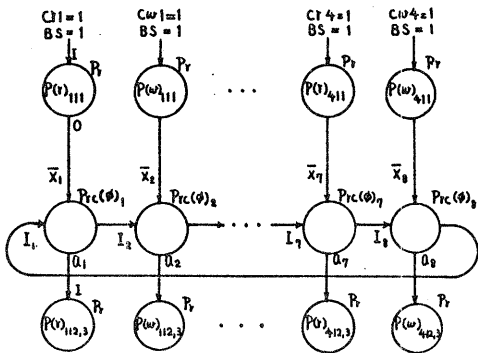


図13. スイッチマトリクス制御の命令構造

非同期回路網であるため、同期問題をハードウェアで解決するための原理図であるフローチャートの各要素の結合関係が、回路網の中でのモジュール間の結合関係と対応しているため、ユーザレベルの記述とハードウェアレベルの記述とが非常に接近している。従って、ACNのシステム設計の年数が少ない(相互排除命令として文献(35)、マルチプロセッサ制御システムとして文献(36)などは比較の1例とする)。

モジュール形式の回路網では、使用されるモジュールのみで相互排除命令が実現できるかどうか回路網の能力判定に大きく係わってくると考えられ、この点からもACNの能力は大きい。さらに、モジュールの種類が極めて少なく2ですむこととこれらの機能表現が単純でよいことから記述が簡単になる点にも長所を持っている。金物の使用量を一層小さくするため、適用例によってはACNとマイクロプログラム化制御回路網との混用が望まれる場合もあろうが、後者の一つ(37)との混用について興味を持っている。なお、制御回路網は一方では回路構造をプログラム可能な論理アレイ(PLA)(38)にする試みがあるが(39)、モジュール型回路網の場合のPLA化は現行の方式では特に無駄が多く(40)新しい方式が望まれる。また同様の意味で、モジュール型回路網のFault-Tolerant化についても一般的なる冗長化構成(41)に頼らない方式が望まれる。

付録 I

〔定義1〕 ACNのプロセスの集合Pの元 P_i が発火するとは、 P_i が状態 S_1 から S_2 へ遷移することをいう。ACNの状態を空プロセスを含ませないすべてのプロセスの第1変数からなるベクトル (e_1, e_2, \dots, e_n) で表わし、その状態集合を Φ とする。さらに、 S_i から S_j ($S_i, S_j \in \Phi$) への状態遷移が P_i の発火によるとき、この状態遷移を $S_i \xrightarrow{P_i} S_j$ と書く。 $w = P_{i_1} P_{i_2} \dots P_{i_n}$ ($P_{i_l} \in P, l = 1, 2, \dots, n$) および状態 S_i, S_k に対して状態の列 S_1, S_2, \dots, S_{n+1} が存在して $S_i = S_1 \xrightarrow{P_{i_1}} S_2 \xrightarrow{P_{i_2}} \dots \xrightarrow{P_{i_n}} S_{n+1} = S_k$ であるとき、 w を S_i から S_k への発火系列と呼ぶ。ACNの発火系列の集合を W_{ACN} で表わす。

〔定義2〕 セーフパトリネットのトランジションの集合及び発火系列の集合をそれぞれT

及び T_{SPN} で表わす。

(定義3) ACNがセーフペトリネットを実現するとは、次の性質(1)及び(2)を持つ $P_0 \rightarrow T^*$ への準同型写像が存在し、かつ(3)が成立することをいう。

(1) f の $P(\leq P)$ への制限は、 $P \rightarrow T$ への全単射である。

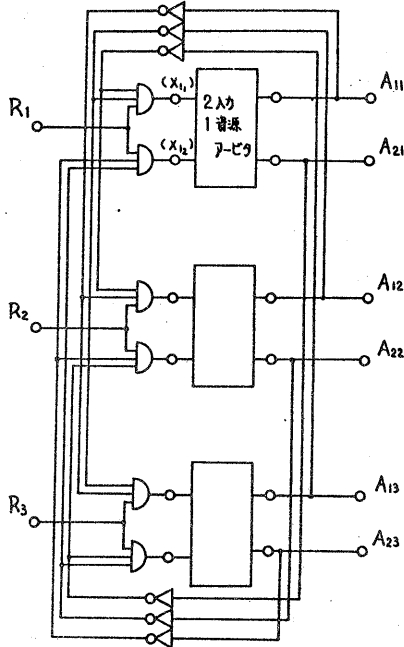
(2) $\forall p \in (P - P')$ なる P に対しては、 $f(p) = \text{入}(空系列)$ とする。

(3) $T_{SPN} = \{ f(w) \mid w \in W_{ACN} \}$ 。

セーフマークグラフを「実現する」という定義も同様に行われる。

付録 II

3入力2資源アービタの論理図を示す。図中の2入力1資源アービタは4・1の方法で構成される。



文献

- (1) Lipton, R.J.: "A comparative study of models of parallel computation", 15th Annual Symposium on Switching and Automata Theory, 1974.
- (2) 辻 例えは, 藤藤: "並列プロセスの制御問題とその応用", 情報処理, Vol.17, 12, P.1169 (昭51-12).
- (3) 例えは, Plummer, W.W.: "Asynchronous arbiters", IEEE Trans. Comput., C-21, P.37 (Jan.1972).
- (4) Husson, S.S.: "Microprogramming-principles and practices", prentice-Hall, INC..
- (5) 明理では, Sholl, H.A. and Yang, S.: "Design of asynchronous sequential networks using read-only memories", IEEE Trans. Comput., C-24, P.195 (Feb. 1975).
- (6) 例えは Bruno, J. and Altman, S.M.: "Theory of asynchronous control networks", IEEE Trans. Comput., C-20, 6 (June 1971).
- (7) Miller, R.E.: "Switching theory, Vol.2, sequential circuits and machines", New York Wiley (1965).
- (8) Shaw, A.C.: "The logical design of operating systems", Prentice-Hall, Inc..
- (9) Courtois, P.J. et al.: "Concurrent control with readers and writers", Comm.ACM, Vol.14, No.10 (Oct. 1971).
- (10) Stucki, M.J. et al.: "Logical design of macro-modules", Spring Joint Computer Conf., 1967.
- (11) Nordman, B.J. and McCormick, B.H.: "Modular asynchronous control design", IEEE Trans. Comput., C-26, 3 (March 1977).
- (12) 中村, 宇都宮: "セミモジュール非同期回路の汎用構成手帳", 信学論(D), 58-D, No.10, P.609 (昭50-10).
- (13) 藤谷 ほか: "中村, 宇都宮 氏の「セミモジュール非同期回路の汎用構成手帳」に対する意見", 信学論(D), 60-D, No.1, P.82 (昭51-1).
- (14) Holt, A. and Commoner, F.: "Events and conditions", Comm.ACM 1970, Concurrent systems and parallel computation, Conference Record.
- (15) Jump, J.R.: "Asynchronous control arrays", IEEE Trans. Comput., C-23, 10 (Oct. 1974).
- (16) 例えは, 野口, 中村: "非同期回路理論の現状", 情報処理 Vol.12, No.10, P.614 (Oct. 1971).
- (17) 塚山 ほか: "非同期順序回路の無ハザード化について", 信学論(D), 60-D, No.6, P.419 (昭52-06).
- (18) Friedman, A.D. and Menon, P.R.: "Systems of asynchronous operating modules", IEEE Trans. Comput., C-20, 1 (Jan. 1971).
- (19) 藤谷: "模塊モジュールによる非同期式順序回路の合成", 信学論(D), 60-D, No.2, P.135 (昭52-02).
- (20) 中村: "非同期回路網の正確性について", 信学論(D), 60-D, No.2, P.175 (昭52-02).
- (21) Misunas, D.: "Petri nets and speed independent design", Comm.ACM, Vol.16, No.8 (August 1973).
- (22) Keller, R.M.: "Towards a theory of universal speed-independent modules", IEEE Trans. Comput., C-23, 1 (Jan. 1974).
- (23) Dennis, J.B.: "Modular, asynchronous control structures for a high performance processor", Comm.ACM Conference record, 1970.
- (24) Patil, S.S.: "Micro control for parallel asynchronous computers", Euromicro, North Holland Publishing Company (1975).
- (25) Gerace, G.B. and Vanneschi, M.: "Flowcharting, microprogramming and system design", Euromicro (1975).

- (26) 西関 その他: "直並列グラフ及びD4ポートの判定法", 信学論(A), 59-A, No.3, P.259 (昭51-03).
- (27) Baer, J.L. et al.: "Legality and other properties of graph models of computations", JACM, No.3 (July 1970).
- (28) Hven, W.H. and Siewiorek, D.P.: "Intermodule protocol for register transfer level modules, representation and analytic tools", The second ann. sympo. on computer architecture, Houston, Texas (Jan. 1975).
- (29) Bell, C.G. and Grason, J.: "The register transfer module design concept", Computer Design (May 1971).
- (30) Commoner, F. et al.: "Marked directed graphs", JCSS, 5, P.511 (1971).
- (31) 柳川, 松原 その他: "μ-バスネットワークによるメモリ共有型ASIC CPU システムASTRAL-1", 信学会計算機研究 EC-70 (1978-2).
- (32) 上論文に関連して, 松原: "バスネットワークのハードウェアによる構成", 情報処理学会, 計算機7-キヲク研究27-1 (1977-7).
- (33) 増山, 吉田: "非同期制御セルによるリニア7-ピタ", 信学論(D), 昭55掲載予定
- (34) Hill, F.J. and Peterson, G.R.: "Digital systems", Hardware organization and design, P.395, John Wiley & Sons, 1973.
- (35) Bredt, T.H.: "Analysis of parallel systems", IEEE Trans. Comput. C-20, 11 (Nov. 1971).
- (36) Moalla, M. et al.: "A design tool for the multilevel description and simulation of systems of interconnected moduls", 3rd Ann. Symp. Computer Architecture (Jan. 1976).
- (37) 増山: "270プログラム制御回路の構成法", 信学論(D), 62-D, No.5, P.357 (昭54-05).
- (38) Fleisher, H. and Maissel, L.I.: "An introduction to array logic", IBM J. Res. & Develop (Mar. 1975).
- (39) 柳川, Jump, J.R. and Fritsche, D.R.: "Microprogrammed arrays", IEEE Trans. Comput., C-21, 9 (Sep. 1972).
- (40) Masuyama, H. et al.: "A programmable logic array realizing asynchronous process network (APN), Part 1 and 2", Memoirs of the Faculty of Engineering Hiroshima University Vol.6, No.3 (Feb. 1978).
- (41) 増山: "マルチトランスポートバス", 信学論(D), 61-D, No.8, P.614 (昭53-08).
- (42) 中村: "非同期回路網の計算と実現", 信学論(D), 59-D, No.8, P.523 (昭51-08).