

セルラー・マイクロプログラム

篠原 健

打浪清一

与塚慶一

大阪大学 工学部

1 まえがき

セルラー・ロジックは、多数の均一な素子をセル状に配列し、セルの動作モードあるいは経線のパラメーターの値を選択する事により、任意の論理機能を実現しようとするものである。現在まで様々なTypeのセルラー・ロジックが提案されている(文献3)が、代表的なものとしては、カント・ポイント・アレイ(文献1)、スリッシュホルド・アレイ(文献2)、又順序関数を実現可能なものとして文献6に示されている方法等をあげることが出来る。更に、最近の集積回路技術の進歩により、プログラマブル・セルラー・ロジックとして商品化されている例もある。

自分自身の中に制御を持てているセルラー・マシンとしては Von-Neuman のもの(文献7)、Holland-マシン(文献8)等が知られている。又ある程度の Control-Structure を持つ、マイクロプログラムの有効なクラスを実現しようとするものには、Microprogrammed-Array(文献9)等がある。又、計算機のサブシステムとして、補助的に、あるいは特殊処理目的に、セルラー構造を用いようとするものは多く、例えば、文献10,11,12等をあげることが出来る。

本稿では、プログラマブル・セルラー・ロジックの1つの方式を提案し、目的とする組合せ、順序関数が与えられて、それをセルのプログラムにまで変換可能な処理系を示す。更に、arithmetic な演算をも含む広範囲の処理を可能にする為、 $m \times m$ の大マトリックスのセル空間を、マクロセルと定義し、これを単位とした構成法をとる。更に、プログラムのフローに対応した制御の実現について考察する。

2 プログラマブル・セルラー・ロジック

プログラム可能セルラー・ロジックは、図1に示す様に同一の構造を持つ、オートマトンを二次元セル状に配列したものである。各セルは複数のモードを持ち、外部から任意のセルを任意のモードにプログラム可能な事ができる。この為には各セルはメモリに於けると同様にアドレスを持つべきであり、指定線 p_r, p_c と書き込み信号線 p を持つ。各セルは4方向の隣接するセルのそれぞれと、2値の信号 $u_i, u_o, d_i, d_o, l_i, l_o, r_i, r_o$ のやりとりを行う。その論理機能をモードに対応して表1の様に定めた。

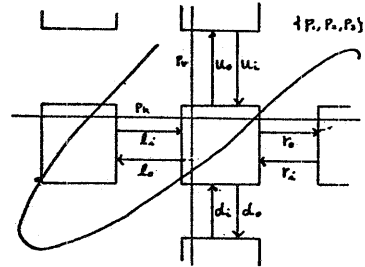


図1 セルラー・ネットワーク

動作	モード	図	u_o	d_o	l_o	r_o
1	静止状態	H_0	<input type="checkbox"/>	0	0	0
2	信号伝送	B_u	<input checked="" type="checkbox"/>	l_i	r_i	u_i
3	“	B_d	<input checked="" type="checkbox"/>	r_i	l_i	d_i
4	交差	C	<input type="checkbox"/>	d_i	u_i	r_i
5	分岐	D	<input type="checkbox"/>	l_i	l_i	0
6	環状	N	<input type="checkbox"/>	0	0	u_i, l_i, d_i

表1 セルのモード

3 記述言語と処理系

この様なセルラー・ロジックで任意の論理機能を実時間内で自動的に生成可能なシステムを考えると、この為には次の3点が解決されるべきである。

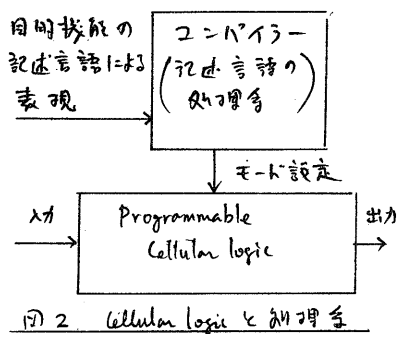


図2 Cellular logic と 処理系

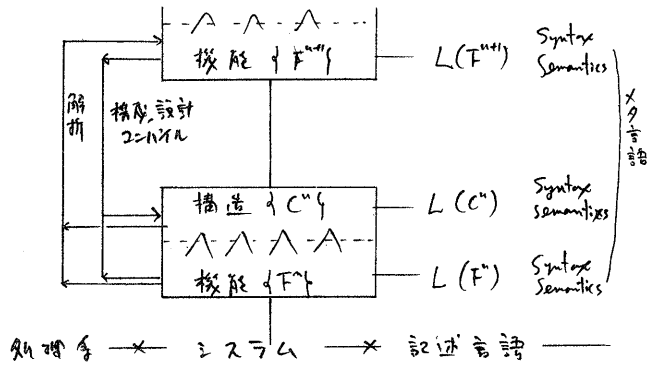


図3 システムの階層性

- i) 論理機能とその組合せ (構造) を明確に表現し得る
記述言語を設計する事。記述言語は
ハードに容着した最も基本的なものが
ハードにこそかゝる。処理の高度な表
現にいたるまで、いくつかのレベルに分
けられ、各々のレベルにおける機能と構
造を表現する記述言語のシンタックスとセマ
ンティクスが共通のメタ言語によって記述
されるべきである。
- ii) i) で対象とする範囲の機能を実現する
為めに基本的に充分な個数のセルの機能と
その組み合わせ方 (構造) を設計する
事。この場合、必ず充分な条件を定め
る事よりも、充分な条件を求め、かつ
対象とする機能が有効にかつ統一的に実現
できる構造と機能を求める事を目標とす
る。
- iii) 目的とする機能を記述したものの外、
記述言語の各レベルにわたって変換して
ゆき、最終的にセル内のモードにまで
変換 (コンパイル) する処理系
を作成する事。

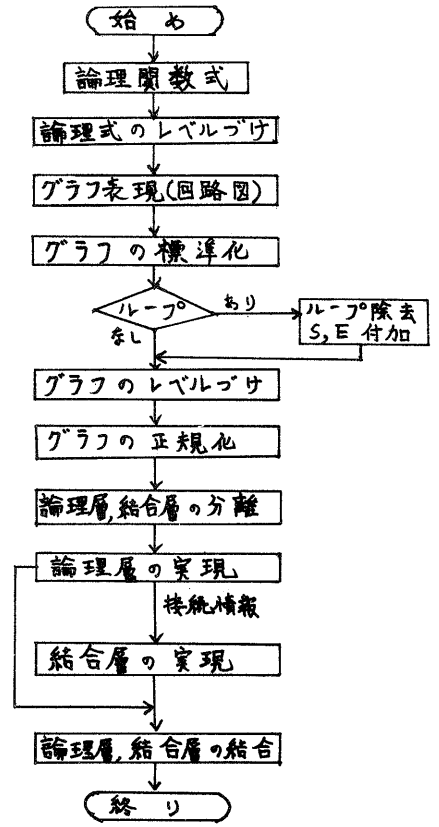


図5 論理関数表現からセルモード表現への変換手順

1. プログラム表現 (ALGOL, APL 等の高レベル表現)
2. 論理関数表現 (組合せ論数 状態方程式など)
3. 回路図表現 (基本素子の 回路図表現)
4. セルモード表現 (セルプログラム)

図4 記述の大きなレベル

4. コンバイラ - 作成実験

図2に示すコンバイラの作成実験を行った。記述言語としては、論理関数表現のレベルを入力とし、セルのモード表現レベルを出力とした。図5にその大ききなフレックサートを示す。又、ニニで各レベルの言語の形式的な定義は省略

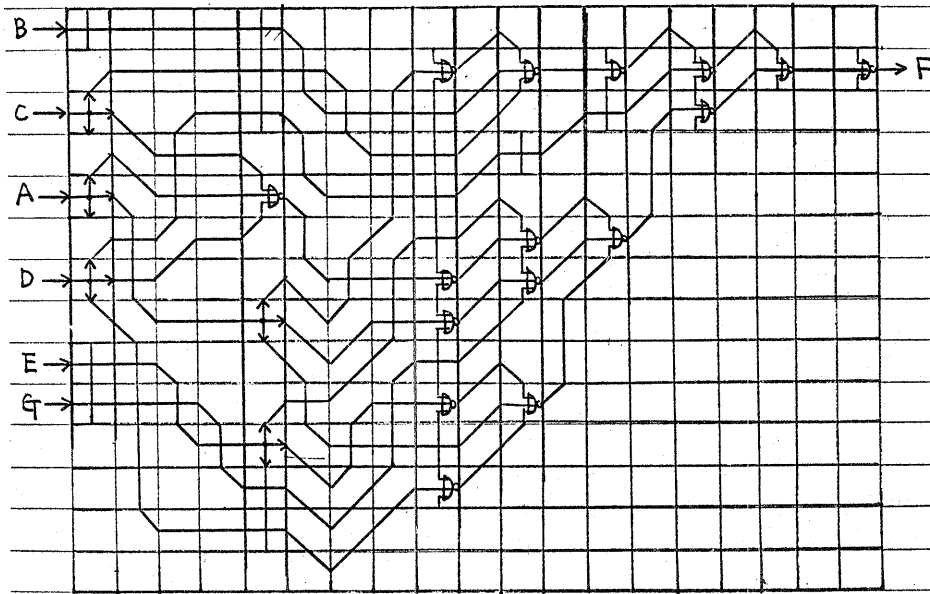


図6. $F = A \cdot \bar{G} + A \cdot D \cdot E + A C \bar{D} \bar{E} + A \bar{B} C D$ に対して
出力されたセルモードプログラム。

す。詳細については文献16を参照。プログラムの大ヨコはFORTRANで約700ステップである。図6に、コンバイルの実行を示す。CPU時間はNEAC 2200/100で約20secであった。

このプログラムでは最適化を行って行なっており、セルモードを決定したものに比して枚数セル数が約、2~3倍程度となっている。

5. 故障診断, 再構成

このセルラ・ネットワークは、実現している論理機能に依存なく一定の順序で、しかも素子数の一次に比例する順序で故障診断が可能である。診断された故障セルは、それと遮蔽する事により、すなわちそのセルを使用しない様に制限を加える事により、正しいセルラ・ネットワークとして使用できる。詳細は文献15を参照。

6. マクロセル

一般の処理では、情報、信号等はビット列であり、これを逐一的に取り扱う為、セル平面を $M \times M$ の大きさの単位に区切り、これを

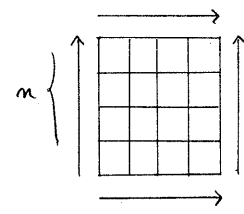


図7 マクロセル

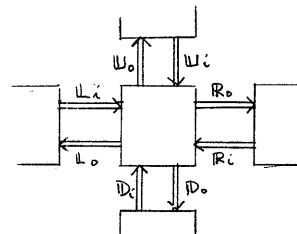


図8 マクロセル空間

マクロセルと呼ぶ。マクロセルは、図8に示す様に隣接する4方向のマクロセルと右値の信号 $U_i, U_o, D_i, D_o, L_i, L_o, R_i, R_o$ のやり取りを行なう事になる。以後、ビットを小文字 x で表現し、ビット列をワードと定義し $X = (x_0, \dots, x_{n-1})$

と表現する。ここで左値というのは*の各ビット \$x_i\$ に \$2^i\$ の重みをつけたものである。重みの大きいビットを上位桁とする時、山, 巾, 川, 川はど丁丁の向きを上位桁にするかで、それぞれ2通りの方法がある。ここでは図7に矢印で示す向きを通常のとりかたとし、それと逆の場合を \$山^1\$, \$巾^1\$, \$川^1\$, \$川^1\$ と表現する。マクロセルをワード処理の基本素子とする場合、次の点が問題となる。

- i) ワード単位の処理において十分な基本機能はなにか。
- ii) ワード単位の基本処理を設定した場合、それをマクロセルで処理可能である際のセルの持つべき機能は何か。
- iii) ワードの書きにかかわらず、一定の文針、手順でセルからマクロセルへの組みあがが可能でなくてはならない。
- iv) セルのポート数は少なくかつその機能はなるべく簡単である事。

7. セルのモードと論理機能

表1の6つのモードに4つ加えて、次の表2に示す10個のモードを定義する。

動作	モード	u_0	d_0	l_0	r_0				
静止状態	H_1		1	1	1	1	OR		0 u_i 0 $l_i + u_i$
分離	T		d_i	r_i	u_i	l_i	4-ト		0 $r_i \cdot u_i$ k l_i
単加算器	A		0	$u_i \cdot l_i$	0	$u_i \cdot o_i$	比較		0
全加算器	F		u_i		l_i		記憶		$\begin{pmatrix} d_0 = r_i \cdot (l_i + \bar{u}_i) \\ l_0 = u_i + \bar{r}_i + \bar{l}_i \\ r_0 = r_i \cdot l_i + u_i \\ d_i = u_i \cdot 0 \\ r_0 = a \\ Q \text{ は } u_i \text{ が } 1 \text{ かつ } r_i \text{ が } 1 \text{ の } l_i \end{pmatrix}$
減算器	S		0	$\bar{l}_i \cdot u_i$	0	$\bar{l}_i \cdot \bar{u}_i$	遅延		d_i 0 $d_0 = C_p$ $(r_0)_n = (l_i)_{n-1}$

表2. 新しく付加するセルのモード

8. ワード単位の処理とマクロセルの構成

ここでは、ワード処理を分類し、そのマクロセルにどのような実装を示す。処理の記述は、アングロ言語に準ずるもの(文献14参照)とする。以下、変数を x, y, z, w で表し、定数を a, A, b, B と表現する。又、マクロセルにおいては処理の単位が常にワードである。セル x はワード $x = (x, 0, 0, 0)$ と等しいものと見做す。処理は大きく分類して、i) 演算(組合せ論的) ii) 情報伝送, 編集, iii) 記憶等のシーケンス処理 iv) 二つの一定の手順に従ってコントロール可能事。

以下、二つの例について述べる。

- I) 演算。二項オペレータで与えられた演算である。二 operand の形が分類し、それに相当するオペレータと、その実装を示す。図に示すマクロセルは $m=3$ の場合であるが、任意の m に對しても構成法

は明らかであらう。

1. 単項演算

1-a) $z = f(x)$ の形 $n \times n$

i) $z = x, z = \bar{x}$

α で示すセルが
 マーカ C の時
 $r_0 = l_i$ として
 $z = x$ を表現、 α がマーカ N
 の時 ($\alpha = N$ を表現する)
 $r_0 = \bar{l}_i$ として $z = \bar{x}$ を表現。

o			
C	C	C	
C	C	C	
α	C	C	
o			

1-b) $z = f(x)$ の形

i) $z = (z_0, z_{n-1})$

$$z_i = x | \bar{x} | 0 | 1$$

右図にあり、対応するビットのマーカを

$\alpha = e | N | H_0 | H_1$ とする事に

する。
 $R_0 = f(l_i)$ として表現される。

B_u	B_u	α
B_u	D	α
D	C	α

1-c) $z = f(x)$

i) $z = +x$

α のビットの論理和

$$r_0 = +l_i$$

ii) $z = *x$

α のビットの論理積

$$r_0 = *l_i$$

iii) $z = \oplus x$

α のビットの排他論理和

$$r_0 = \oplus l_i$$

α			
C	C	C	
C	C	C	
P	P	P	$*x$

B_u	C	C
A	C	C
A	C	C

$*x$			
C	C	C	
C	C	C	
B_u	A	A	$\oplus x$

1-d) $z = f(x)$

i) $z = \text{perm}(x)$

任意の置換

$$\alpha = C | B_u$$

$$R_0 = p_1(l_i), W_i = p_1^{-1}(R_i)$$

$$D_0 = p_2(l_i), U_i = p_2^{-1}(D_i)$$

あるいは

$$\alpha = C | B_u$$

$$W_0 = p_1(l_i), U_i = p_1^{-1}(W_i)$$

$$R_0 = p_2(D_i), D_i = p_2^{-1}(R_i)$$

B_u	α	α
α	B_u	α
α	α	B_u

α	α	B_u
α	B_u	α
B_u	α	α

ii) $z = (z_0, z_{n-1}) = f(x)$

$$z_i = x_j \quad 0 \leq i, j \leq n-1$$

任意の置換

$$\alpha = D | C$$

(ただし、D は各列に 1 つ)

$$W_0 = D_0 = f(l_i)$$

$$R_0 = l_i$$

iii) $z = \text{mask}(x)$

$$z_i = x_i | 0 | 1$$

$$\alpha = C | H_0 | H_1$$

$$R_0 = \text{mask}(l_i)$$

$$U_0 = \text{mask}(R_i)$$

あるいは

$$\alpha = C | H_0 | H_1$$

$$D_0 = \text{mask}(W_i)$$

$$W_0 = \text{mask}(D_i)$$

iv) $z = \bar{x}$

$$R_0 = \bar{l}_i$$

α	α	α
α	α	α
α	α	α

α	C	C
α	C	C
α	C	C

C	C	C
C	C	C
α	α	α

N	C	C
N	C	C
N	C	C

2. 二項演算

2-a) $z = f(x, y)$

i) $z = x \cdot y$

$$r_0 = l_i \cdot u_i$$

ii) $z = x \oplus y$

$$r_0 = l_i \oplus u_i$$

iii) $z = x + y$

$$r_0 = l_i + u_i$$

C	C	C
C	C	C
A	C	C

C	C	C
C	C	C
P	C	C

2-b) $z = f(x, y)$

i) $z = f(x, y)$

$$z_i = x | y | 0 | 1$$

$$\alpha = H_0 | H_1 | C | B_u$$

$$R_0 = f(x, y)$$

または

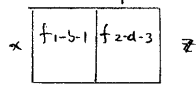
$$D_0 = f(x, y)$$

α	α	α
α	α	α
α	α	α

2-c) $Z = f(X, Y)$

i) $X=1$ の時 $Z=Y$
 $X=0$ の時 $Z=0$

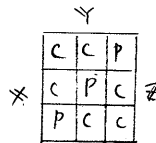
これは $(X-X \cdot X) \cdot Y$ と等しく、
 従って 2-d-iii) の $X \cdot Y$ と 1-b-i)
 の $X = f(x)$ の合成として実現でき



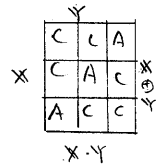
2-d) $Z = f(X, Y)$

これは logical 操作 A と Arithmetic
 操作 B に分解できる。可也。
 Logical 操作 A をあげると、

i) $Z = X + Y$
 ビット毎の OR
 $R_0 = \cup_i \cup_i$



ii) $Z = X \oplus Y$
 ビット毎の Ex-OR
 $R_0 = \cup_i \oplus \cup_i$

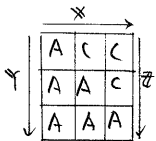


iii) $Z = X \cdot Y$
 ビット毎の AND
 $R_0 = \cup_i \cdot \cup_i$

Arithmetic 操作 A をあげると、
 たゞし 2^{-1} 以下の大元の自然数
 を対象とする。整数、実数への
 拡張は、その表現方法を定めれば
 可能である。1) 可換自然数
 の上での算術を基盤にして拡張して
 ゆく事ができる。

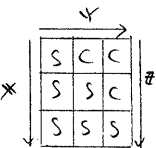
iv) $Z = X \text{ odd } Y$

加算
 $R_0^{-1} = \cup_i \text{ odd } \cup_i^{-1}$
 Overflow は
 $X = +D_0$ で得られる。



v) $Z = X \text{ sub } Y$

引算
 $R_0^{-1} = \cup_i^{-1} \text{ sub } \cup_i$
 同時に桁借りは
 $C = +D_0$ で得られる



vi) $Z = \max(X, Y)$

X, Y を算術的に比較し大きい
 の方を取り出す。



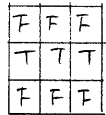
$R_0 = \max(\cup_i, \cup_i)$

vii) $Z = X \text{ multi } Y$

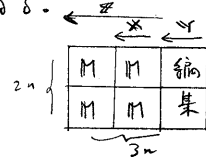
これを1つのコクロセルで実現する
 事はできない。 $Z = Z$

$X = (x_0, \dots, x_{n-1})$ に初して
 $X' = (x_0, 0, x_1, 0, x_2, \dots)$ を定義
 した時

$\cup_0^{-1} = R_0^{-1} \text{ multi } \cup_0^{-1}$ を
 右図で実現できる。



= のコクロセル M を用い
 M ビット \times M ビットの掛算器
 は次の様に作る。



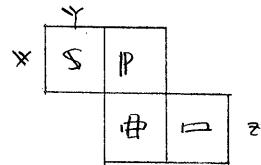
2-e) $Z = f(X, Y)$

これは relation と作る。 Z 作る。

i) $X = Y$

$(X = Y) ::= - + (X \text{ sub } Y)$

であるから。 ($::=$ は \times の記号)
 対象可逆コクロセルを接続して、

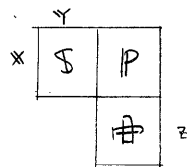


と実現できる。

ii) $X \neq Y$

$(X \neq Y) ::= + (X \text{ sub } Y)$

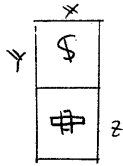
故に、



iii) $X > Y$

$$(X > Y) ::= +(\text{Carry of } Y \text{ sub } X)$$

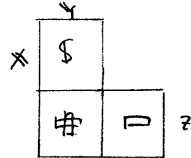
故に



iv) $X \geq Y$

$$(X \geq Y) ::= -(Y > X)$$

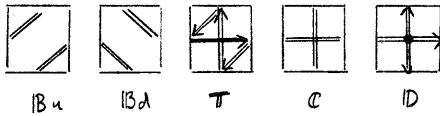
故に



II] 情報伝送, 編集

$$X := Y$$

次に示すマクロセルは、セルのレベルと全く同じ取り扱いかたがで可。従って、あるマクロセルから、他のマクロセルへ情報を伝送する様に全体を配置する事は常に可能である。



又、ワードの編集については、

i) サブワードの抽出

$$\begin{aligned} \text{例) } X(1,3,6) &= (x_1, x_3, x_6) \\ X(2-4) &= (x_2, x_3, x_4) \end{aligned}$$

ii) カスケードワードの合成

$$\text{例) } X(1,3) \text{ 及 } Y(2,4) = (x_1, x_3, y_2, y_4)$$

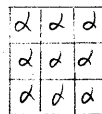
この2つは、次に示す様なマクロセルを用いたは実現可能である。

$$d = C | B_u | B_d$$

$$(d_1, d_2, d_3, d_{n-1})$$

$$= (u_1 \dots u_j, r_1 \dots r_k, d_m \dots d_n)$$

$$1 < j < k < l < n < m'$$

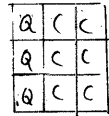


これは、マクロセルの3つの面を順に接続した長方形のセル、行列としての1つの面とを順序を得、2抽出出来る事である。

III] 記憶: i) $U_i \Rightarrow Q$

・レジスタマクロセル

は右図の様である。



U_i の立ち上がりを検出して

その時の U_i を Q にする。

$$d_0 = U_0$$

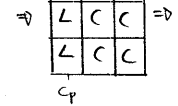
$$Q \leftarrow U_i$$

$$R_0 = Q$$

ii) 又、各部を同期的に作動させる為、内部クロックに同期した T_F

$$(R_0)_k = (U_i)_{k-1}$$

$$d_0 = C_p$$



これは L の値を遅可事により

$(R_0)_k = (U_i)_{k-1}$ と変更可能。

$$0 \leq i \leq n$$

9. マクロセル・プログラミング

マクロセルの機能は、Word 単位の処理の基を考えたものである。Word 単位の全部をこなしている。また示した様に、マクロセルを二次元的に組み合わせる事により、より大規模なインストラクションも実現可能である。

$$\text{例) } \text{If } X \geq 0 \text{ then } Z = X \text{ sub } Y \\ \text{else } Z = X \text{ odd } Y$$

をマクロセル・プログラムに変換可能。

これは

$$T = \text{bitfall}(- + (\text{Carry of } 0 \text{ sub } X))$$

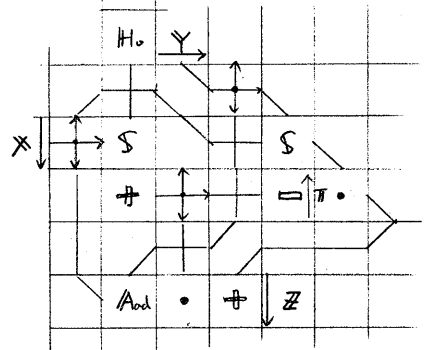
$$Z = T \cdot (X \text{ sub } Y) + (-T) \cdot (X \text{ odd } Y)$$

である事から、2 ID のマクロセル・プログラムを得る。

又、 Δ , Π 号を使用して
順序回路を合成可能であり、 Φ と
隣接な処理も可能である。

10. コントロールの実現

上に示した指令 処理のブロックは、
プログラムで言えば、1つのインストラクション
に相当するものである。プログラムは、
基本処理を記述したフローチャートで表わす
れる。従って、有数のプログラムを表現
する際には、フローチャートに対応して
例えば、 $Z = X \cdot Y$ 、判断による分岐等
に対応する制御が可能でなければならぬ。
これは、⑩11に示す様に、セル内部に
コントロールロジックを表現し、マクロ
セル内部の各ブロック間の情報を基にして、各ブロックを適宜活性化す
る事に依り実現できる。

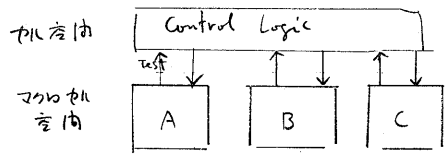


⑩10 If $X \geq 0$ then $Z = X \text{ sub } Y$
else $Z = X \text{ add } Y$
に相当する
マクロセルプログラム。

11. あとがき

プログラマブルセルロージックを用いて、
arithmetic な演算をも含む、広リクスの物理
を行なう実現に依りて考察した。

これは、従来のセルモードを各種類
領域をマクロセルと定義し、これをMセルのポートを取り扱う基本セル
と考へた。マクロセルを単位としたマクロセル空間は、基本セル空間と同じ
構造のセル空間となり、統一の取り扱いが可能である。単一のマクロセル
の果ては機能は、加減算をも含む、通常のワード単位の処理を旨と考へるもの
である。本稿では、実際に、これらの演算に依りマクロセルの設計を行な
た。これは一般にMセルの大きさに依りなく、同一の機能に対しては、同一の台帳
で構成できるものであった。又により高い機能を実現する為には、マクロセル
を2次元的に接続してゆく必要がある。これは、マクロセル内部でのモード設定
(プログラム)であり、統一の語を考へる事が可能である。更に、制御
機能を内部にもつて実現も可能である。取組んだ内部としては、より高層な
機能の記述言語を設定し、これをマクロセルプログラムに変換し、セルのモード
を設定してゆく処理系を作成する事である。又、この様に、内部的に分布した
処理機能を持つシステムは、本質的に並列処理の能力を持つ、と考へる。特に、本
稿で述べてきた様に、データの伝送路が、常に確保されているタイプでは、そう
であり、並列処理の能力を有効に引き出す為のプログラム・インターフェースの
作り方を検討してゆく必要がある。



⑩11 制御を持ったプログラム

最後に、同項熱心に討論して下さった研究室の諸氏、特にコンパイラーの
作成実験として下さった西野氏に謝意を表します。

参考文献

1. Cut-point Cellular Logic
R.C. MINNIC IEEE TC Dec '64 pp 685~698
2. A cellular threshold array
W.H. Kautz IEEE TC EC-16 '67 pp 680~682
3. A survey of Microcellular Research
ROBERT C. MINNIC JACM vol 14 No.2 Apr. '67 pp 203~241
4. An Augmented Content-Addressed Memory Array for Implementation with Large-scale Integration
W.H. Kautz JACM vol 18 No.1 Jan. '71 pp 19~33
5. A Universal Cellular Array
Jung - CHANG - HUNG IEEE TC MARCH '71 pp 317~320
6. Cellular Synthesis of Synchronous Sequential Machines
SUNG C. HU IEEE TC Dec. '72 pp 1337~1405
7. Theory of Self-Reproducing Automata
Von Neuman Univ of Illinois Press '66
8. Iterative Circuit Computers
Holland Proc. 1960. Western Joint Comp. Conf. 395~400
9. Microprogrammed Arrays
J.R. JUMP, DR FRITSCHER IEEE Vol C-21 No.9 Sep '72
pp 974~984
10. System Design of Cellular APL Computer
K.J. THURBER, J.W. MYRNA, IEEE Vol C-19 No.4 April '70
pp 291~303
11. A Flow Mode, Self Steering Cellular Multiplication - Summation Processor
D. BJORNBERG BIT 10 '70 p 125~144
12. Cellular Arrays for the solution of Graph problems
K.N. LEVITT W.H. Kautz CACM. Vol 15 No.9 Sep. '72 pp 889~901
13. RECENT DEVELOPMENT IN SWITCHING THEORY
MUKHOPADHYAY ACADEMIC PRESS 1971
14. An ALGOL-Like Computer Design Language
Yaohan Chen. CACM vol 8 No. 10 Oct '65 pp 607~615
15. 不能論理回路とその構成セル, その設計手順について
保存, 打取, 自取 電学協研誌 1972-3
16. 記述言語によるハードウェアの自動構成
保存 他. 電学協研誌 1972-7
17. Failure Operable な不能論理回路とその構成セルについて. 1972 全大
保存 他
18. セル状可変論理回路を用いたハードウェアについて
保存 他. 電学協研誌 1973-1