

# Zuse Z1 の演算機構

和田 英一 (IIJ 技術研究所)

wada@u-tokyo.ac.jp

Konrad Zuse がプログラマブルな機械式 2 進法計算機を試作したのは 1936 年頃といわれている [0]. Zuse 関係の文献は整理されていて, インターネットで取ることが出来る [1]. 機械式であったとか, 2 進浮動小数点方式であったとか, プログラムはフィルムに空けた穴で表わすとか, ほぼ同様なアーキテクチャでリレー制御の Z3 には平方根命令があったとかは判るが, 肝心の演算機構は定かではなかった. やっと別の文献 [2] にいろいろ書いてあることが判明した. しかし, どうもこれを書いた人達も精通しているとは思われない節もある.

以下では私流の解釈で, Z1 の演算機構を説明したい.

## 論理演算機構

素子は左の図 0 のように金属の板に空けた穴に棒が差し込める構造である. 金属板は離ればなれに描いてあるが, 実際は層状に重なっている. 次の図 1 で名称を紹介するが, 上から能動板, 受動板, 制御板といい, 図ではリレーでいえば電流が流れていない時 (定位) を示す. それぞれ矢印の方向に動くことができる.

リレーでは, 節点を閉じればすぐ電流が流れるが, 機械回路ではそうはいかない. 節点を閉じ, 回路の一端に電圧をかけるともう一方の端に電圧が出るような仕掛けである. 電圧をかけるのに相当するのが, 能動板を x 方向に矢印分動かすことで, この図では能動板を動かしても棒の位置には影響を与えず, 従って受動板は動かない. 最下段の制御板を y 方向に動かすと, 棒も y 方向へ矢印分動き, その状態で能動板を x 方向へ動かすと, 棒も x 方向へ動き, 受動板も x 方向へ動く. これが基本である.

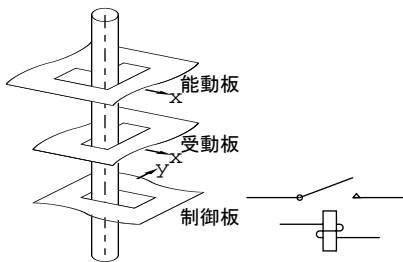


図 0 基本素子

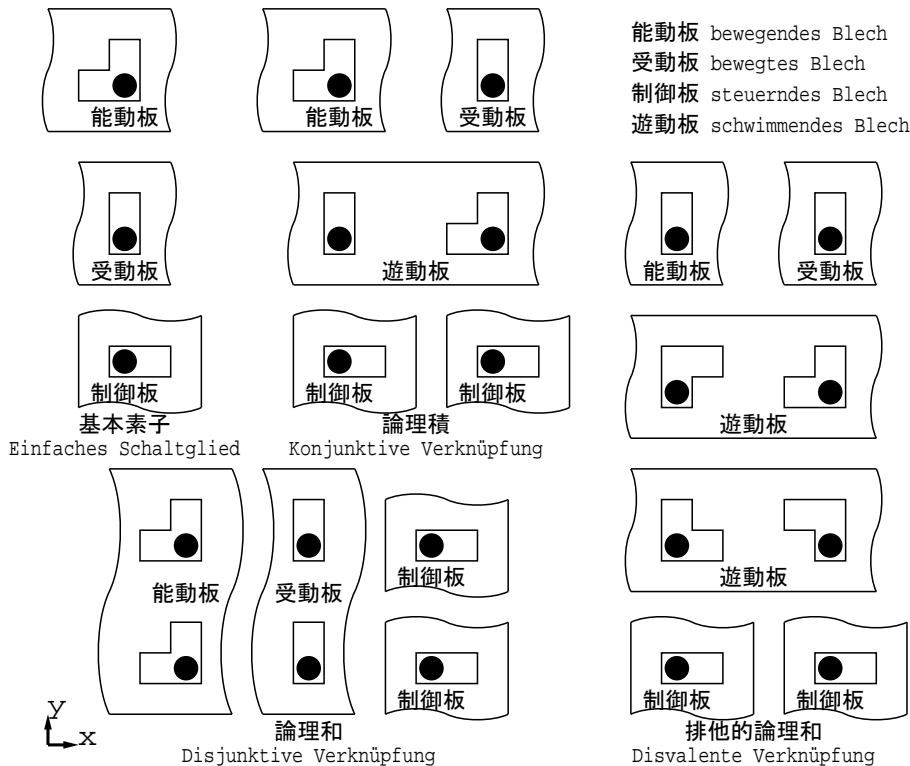


図 1 論理演算機構

金属板を使った機械式論理回路を図1で説明する。左上が基本素子で図0と同じものである。各板の相対位置は棒の位置から判断出来る。このまま能動板が右(x方向)に動いても、(黒丸で示す)棒の左の空間のため棒は動かず、従って受動板も動かない。制御板が上(y方向)に動くと、棒はそれに押されて上に移動する。すると能動板では棒の左の空間がなくなるので、能動板が右に動けば棒も右に動き、受動板も右に動く。

基本素子の下が論理和で、右方に制御板が2個ある。このままで能動板を右に動かしても、受動板には影響はないが、制御板のいずれかを上に動かせば、能動板ではいずれかの棒の左方の空間がなくなり、棒を右方に押し遣り、受動板が動く。

基本素子の右隣りが論理積である。ここでは遊動板を1枚使う。下方の制御板がともに上に動いていれば、棒はともに上に移動して、能動板が動けば遊動板も動き、遊動板が動けば受動板も動く仕掛けである。

一種独特な切掛けの遊動板を2枚用意すると、排他的論理和が実現出来る。制御板をこのままにしておき、能動板を右に動かすと、左の棒は右に動く。従って上の遊動板は右に動くが、下の遊動板は動かない。いずれにしても右の棒を動かすことは出来ず、受動板は動かない。

次に左の制御板が上に動いたとすると、左の棒も上に動き、能動板を右に動かすと下の遊動板が機能して右の棒も動き、受動板も右に動く。また右の制御板が上に動いたとすると、上の遊動板が機能して右の棒が動き、受動板も右に動く仕掛けである。両方の制御板が上の位置にあるときは、最初と同じで受動板を動かすことは出来ない。

論理演算にはこの他に否定があるが、梃子の理で押し引きを逆にすればよい。

### 桁あげ予測回路

Z1は2進法の浮動小数点演算回路を持っていたといわれる。演算の基本は加算なので、それを考える。

まず  $(a_7a_6\dots a_0)_2$  と  $(b_7b_6\dots b_0)_2$  を各桁ごとに足して  $(c_7c_6\dots c_0)_2$  を得る。次に下の桁からの繰上がりだが、 $c_0$  には加算の時は繰上がりは来ない。 $a_0$  と  $b_0$  がともに1のとき、 $c_1$  に繰上がりがある。この繰上がりは  $c_1$  に影響する他、 $a_1\dots$  と  $b_1\dots$  が0と1の組である限り上に伝わっていく。 $a_1\dots$  と  $b_1\dots$  が0,0か1,1になると止る。それを見ているのが  $c_i$  で開閉するスイッチの絵である。

最左端のANDは  $c_7$  のスイッチから来る繰上がりとともに、この加算のオーバーフローを定める。

最右端の  $c_0$  のスイッチは減算の時、1の補数の加算に加える繰上りを注入する口である。減算の時は  $b_i$  を上のXOR, ANDに入れる前に補数にする。従ってもう1段のXOR回路が必要である。

図2のXOR, ANDの入力は図1の制御板に対応している。機械式計算回路を構成するときは、制御板入力に確定した時点で、能動板を動かすタイミングが必要になる。それにより受動板から出力が得られる。

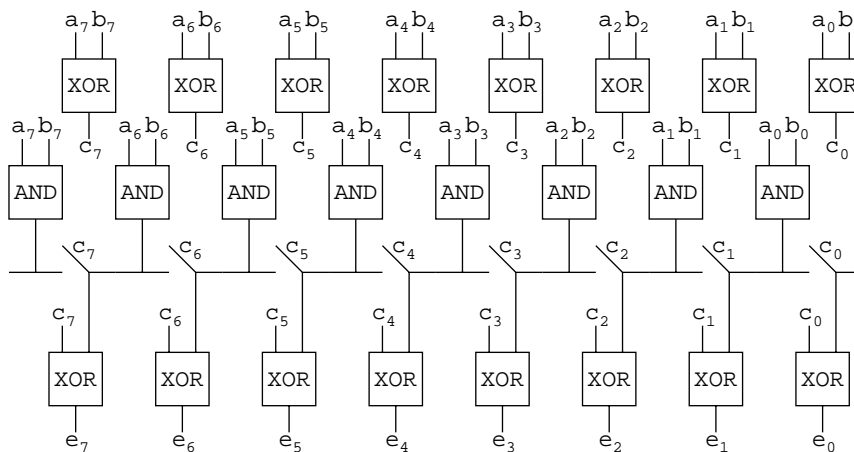


図2 加算回路

### 記憶装置

記憶装置も大変ユニークであった。図3の左端は22ビット64語の記憶装置の始めの方を示す。各語は縦方向のビット列で構成され、それらが語0から語1, 語2, ... と横方向に並ぶ。

図3で右へ順に分解した部品を示す。最初が固定板(c)で、「凹」の字を上下逆したような穴が開いていて、そこに棒を差す。その上に各語に対応した縦向き細胞板(a)が来る。更にその上(かどうかは関係ないが)に各ビットに対応した横向きの制御板(b)が乗る。

語を選択するときは、それに対応した a の板を動作させる。

...

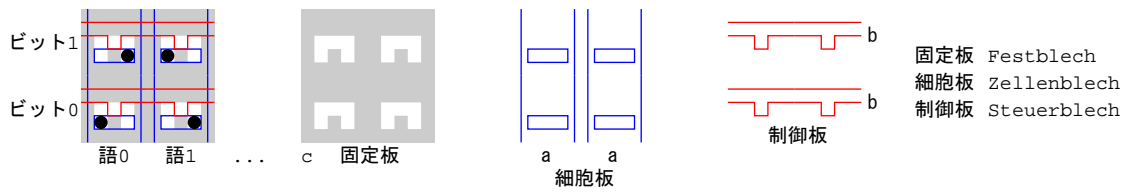


図3 記憶装置

読出しの手順は次の通り (図4). 左から順に操作する. まず読み出す語の a の板を上へ動かす. すると棒は 0 にあれば 0' へ, 1 にあれば 1' へ移動する. a は制御板として働く. 次にその a を右へ動かす. 0' にある棒は b の突起に当り, b を右へ動かす. また 1' にあると, a により右へ押されることはなく, 突起には当らず b は動かない. この a は能動板といえる. b の右方への動く, 動かないで対応するビットが 0 か 1 が判明する. 従って b は受動板に相当するといえる.

図には示さないが, 読み終わったら 0 の場合は a を左へ戻し, b を左へ戻し, a を下へ戻す. 1 の場合は a を左へ戻し, 下へもどす.

書込みは図5に示す. 書く前にその記憶が 0 か 1 かは判らないので, 始めの棒の位置は示してない. 0 を書くビットは b を右へ 2 単位動かし, 1 を書くビットは b を左へ 2 単位動かしておく. 次に a を上へ動かす. b をもとへ戻すと棒は 0 を書くか 1 を書くかにより 0' か 1' に来る. そこで a を下へ動かすと棒は希望通り, 0 か 1 の位置に収まる.

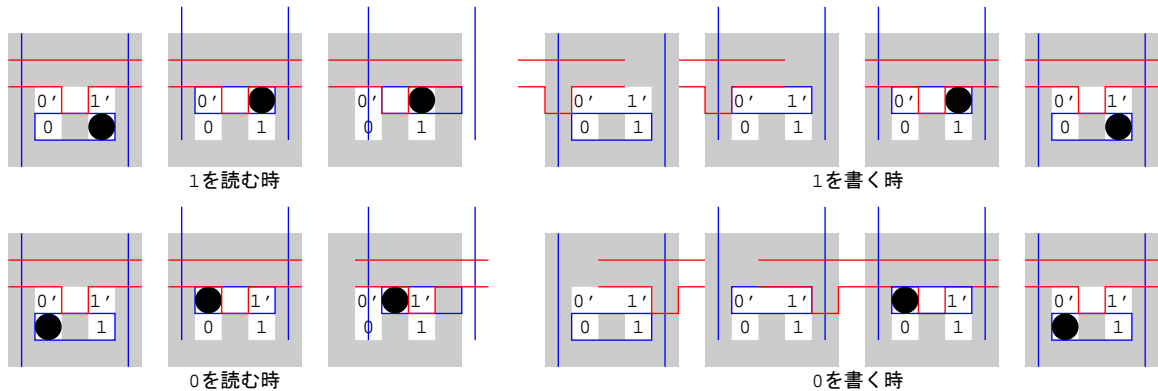


図4 記憶読出し

図5 記憶書込み

参考文献

[0] Raúl Rojas: "Konrad Zuse's Legacy: The Architecture of the Z1 and Z3," IEEE Annals of the History of Computing, Vol.19, No.2,1997,pp.5-16.  
 [1] <http://www.zib.de/zuse/English.Version/index.html>  
 [2] U. Schweier & D. Saupé: "Funktionen und Konstruktionsprinzipien der programmgesteuerten mechanischen Rechenmaschine Z1," Arbeitspapiere der GMD 321, Bonn, 1988.