技術解説
14-A の計算アルゴリズム

はじめに

14-A の商品化は 1957 年。開発には 7 年を要しました。当時はトランジスタが誕生したばかりでまだ IC やマイコンは世に無く、入手可能な電子素子は真空管やリレー素子や高価なトランジスタなどに限られていました。

14-A の回路はリレー素子を用いて、スイッチ素子間を繋ぐ膨大な配線網で演算機能を実現しています。その回路内にはプログラム制御がなく、ロジックゲート回路さえも使用されていません。現在、四則演算機能はパソコン上のプログラム言語でいとも容易に作成できますが、14-A では、60 年前の限られた技術環境で創意工夫を凝らしたアルゴリズム設計により、コストと性能を両立させた計算機を実現させています。

発明家・樫尾俊雄の残したこの技術を後世に伝えるため、本資料を作成しました。

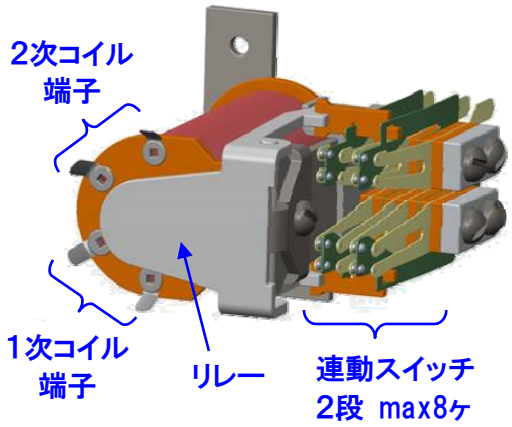
2013 年 6 月
カシオ計算機株式会社　代表取締役副社長　樫尾　幸雄

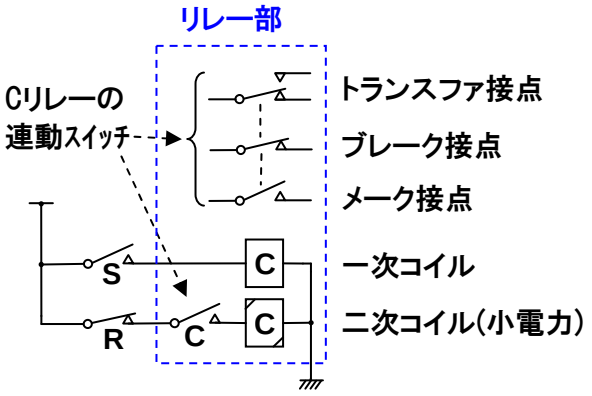
目次

1. リレー素子

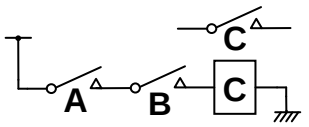
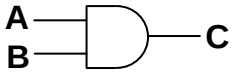
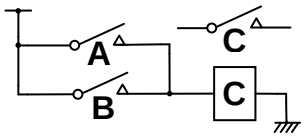
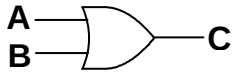
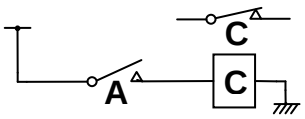
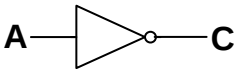
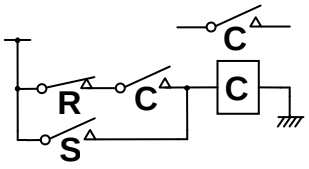
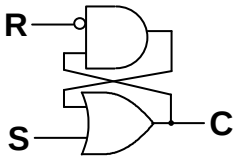
当時、リレーは電話交換機の通話接続スイッチとして多用されていた。そのリレーが論理回路を構成できることに着眼して計算機を設計している。下図のようにリレー素子の組み合せで、AND、OR、NOT などの論理回路と情報を蓄積するラッチ回路が実現できる。論理回路とラッチ回路（メモリー）が構築できれば、原理的に計算機の開発は可能である。商品化にはコスト低減とオフィス設置スペースへの工夫が必要で、高価なリレー部品の数を 341 個に抑えている。大型コンピュータでは 10000 個以上が使われていたようで、14A のリレー個数の少なさは画期的であった。またリレー素子は、計算回路に適するよう工夫改良を加えた独自設計となっている。

リレー部品





リレー回路と論理

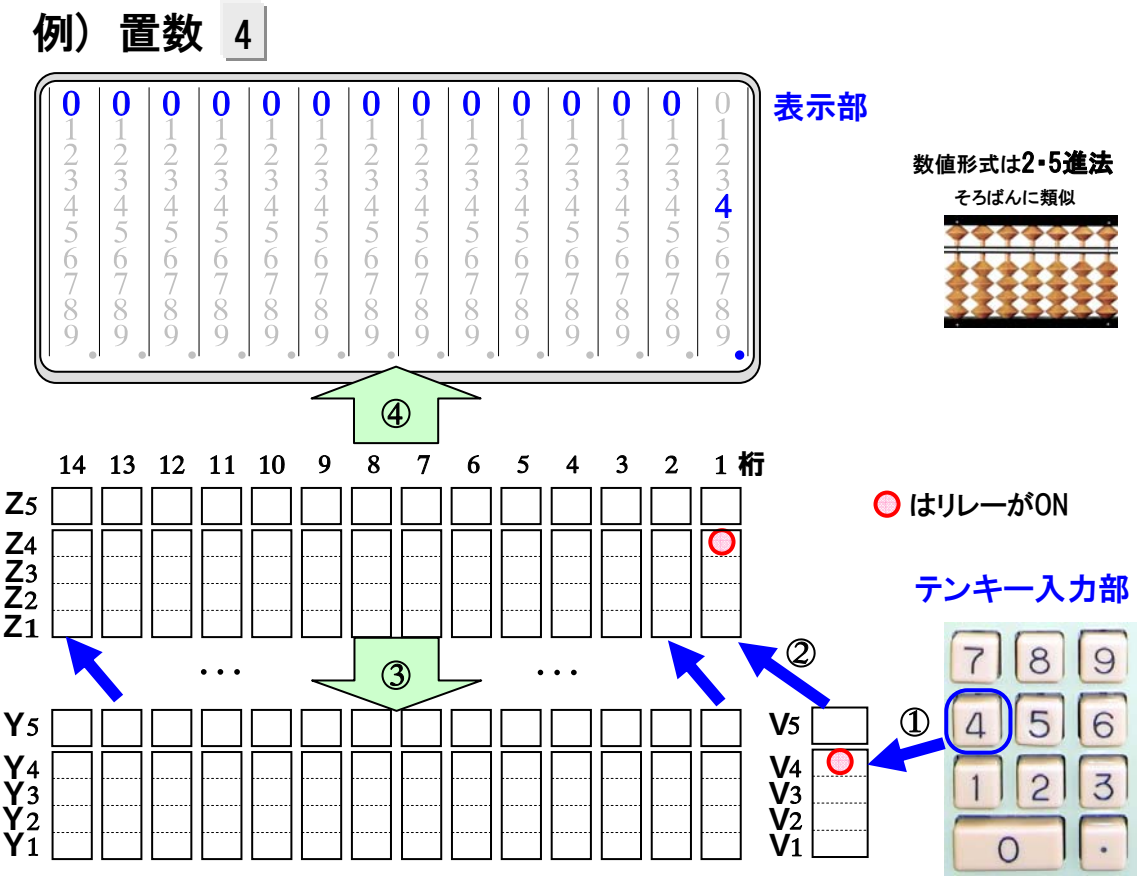
AND		
OR		
NOT		
Latch		

2. 置数制御

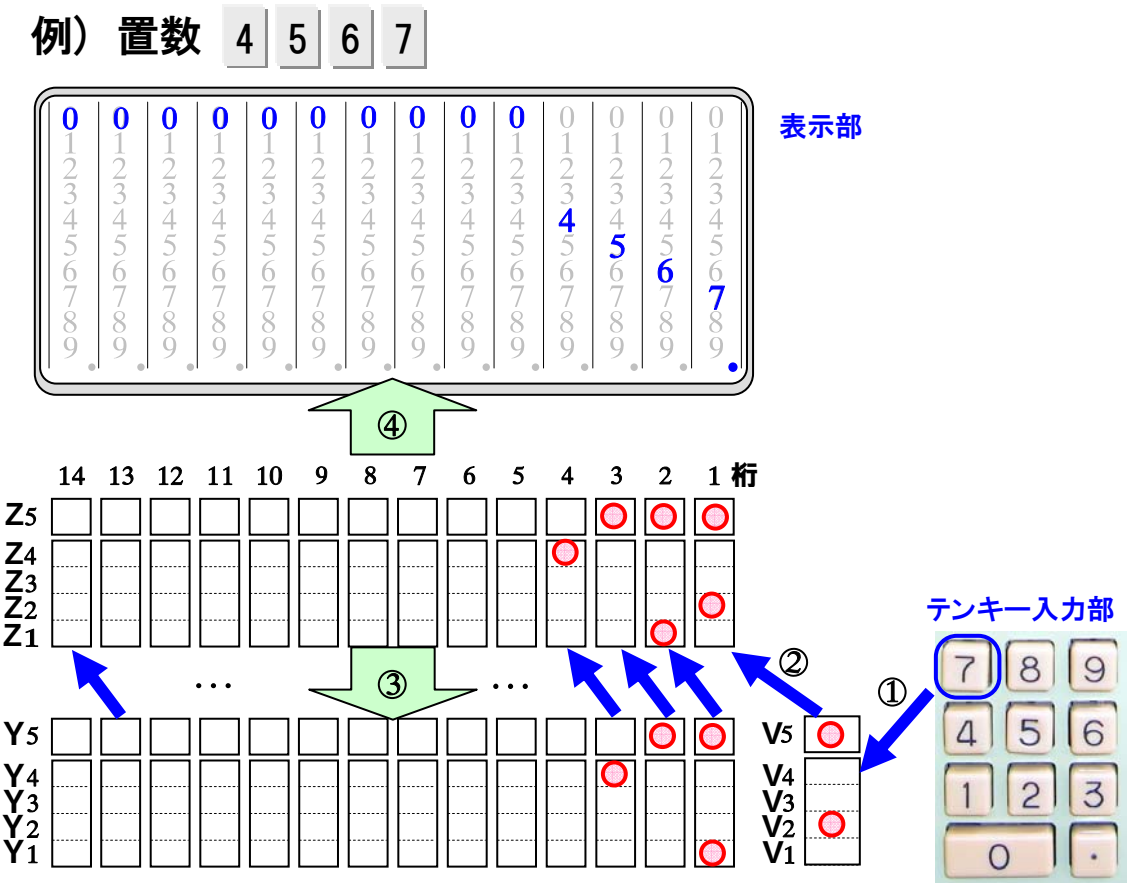
当時の計算機は各桁毎に 0~9 のテンキーがマトリックス的に割り振られていた。リレー計算機 14-A では、現在の電卓テンキーの原型となる 1 組のテンキーによる置数入力が開発された。テンキーからの連続入力で多桁の数値入力が可能となった。現在では多くの電子端末にあたりまえのように搭載されているが、この機能の実現は 14-A が初めてであった。では、この置数制御のしくみを追跡してみる。

表示部は各桁毎に 0~9 の数値ランプと小数点ランプから構成している。テンキー入力部から数値キーが操作されるとリレーV1～V5 にその数値が保持される。数値形式はそろばんの 5 玉に類似した 1 桁にラッチ 5 個で構成された 2・5 進法である。例えば、数値 4 は V4 リレーのみ ON、数値 7 は V5 と V2 リレーがＯＮとなる。また、14 桁数値を保持する 3 本のレジスタ X、Y、Z を駆使して、置数や各種演算が実施される。

では電源ＯＮ後に、数値 4 キーを操作した場合を追跡する。数値 4 入力でラッチ V4 がＯＮとなり、その情報はレジスタ Z の 1 桁目にコピーされ、1 桁目の Z4 が ON となる。またレジスタ Z の 2 桁目は、レジスタ Y の 1 桁目からコピーされる。同様処理にて、レジスタ Z の 3~14 桁目にはレジスタ Y の 2~13 桁目がコピーされていく。すなわちはラッチＶを介した置数数値とレジスタ Y の桁上げ数値がレジスタ Z で合成されたことになる。そして桁上げされたレジスタ Z の情報はレジスタ Y にコピーで戻される。またレジスタ Z は表示部に導出され、表示部各桁の該当数値と小数点が点灯される。



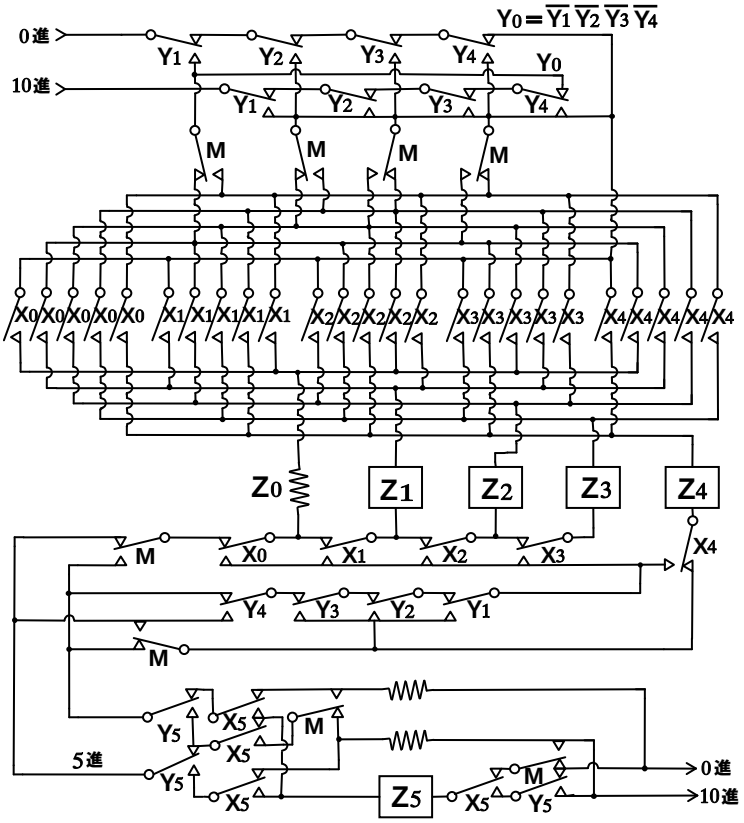
続いてテンキーより 4 に続いて 5、6，7 と置数入力される場合を図示する。一式のテンキーからの同様な連続置数操作により、14 桁までの数値入力が可能となる。



参考）14－A の操作パネル

4. 演算回路

2・5 進法による演算回路を説明する。下図は 1 桁分の演算回路で、14 個の演算回路が直列に繋がって、14 桁全体の演算回路となる。



1 桁の演算回路　X±Y⇒Z

M=0(ブレーク状態)は加算回路で、M=1(メーク状態)で減算回路に切り替わる。また M リレーは各桁独立で加減算を制御できる。演算回路は 1 桁から 14 桁まで通電して初めてリレーが動作する。

Z1~Z4 リレーからの選択が 5 進演算結果の 1~4 となり、非選択の結果 0 (Z0) は抵抗での代用となる。また、5 進のキャリー有り（5 進信号）か無しかが選択され、そろばんの 5 玉に相当する 2 進演算部に送られる。2 進の Z5 リレーが通電すると+5 で、非通電が+0 の状態を示す。Z1~4 と Z5 の状態の組み合わせで、数値 0~9 を表わす。また、2 進演算結果によるキャリーの有りと無しは、それぞれ 10 進と 0 進の信号として次桁の演算回路に送られる。

減算での 5 進 2 進の各ボロー信号は加算でのキャリー信号と同じ信号となる。左図の「2・5 進法演算の論理」には、この演算回路の全ての論理が掲載してある。

M＝0　X＋Y＝Z

X＼Y	0進 Y0	Y1	Y2	Y3	Y4	10進 Y0	Y1	Y2	Y3	Y4
X0	Z0	Z1	Z2	Z3	Z4	Z1	Z2	Z3	Z4	Z0
X1	Z1	Z2	Z3	Z4	Z0	Z2	Z3	Z4	Z0	Z1
X2	Z2	Z3	Z4	Z0	Z1	Z3	Z4	Z0	Z1	Z2
X3	Z3	Z4	Z0	Z1	Z2	Z4	Z0	Z1	Z2	Z3
X4	Z4	Z0	Z1	Z2	Z3	Z0	Z1	Z2	Z3	Z4

X＼Y	0進 Y5	Y5	5進 Y5	Y5
X5	Z5	Z5	Z5	Z5
X5	Z5	Z5	Z5	Z5

M＝1　X－Y＝Z

X＼Y	0進 Y0	Y1	Y2	Y3	Y4	10進 Y0	Y1	Y2	Y3	Y4
X0	Z0	Z4	Z3	Z2	Z1	Z4	Z3	Z2	Z1	Z0
X1	Z1	Z0	Z4	Z3	Z2	Z0	Z4	Z3	Z2	Z1
X2	Z2	Z1	Z0	Z4	Z3	Z1	Z0	Z4	Z3	Z2
X3	Z3	Z2	Z1	Z0	Z4	Z2	Z1	Z0	Z4	Z3
X4	Z4	Z3	Z2	Z1	Z0	Z3	Z2	Z1	Z0	Z4

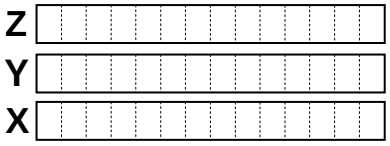
X＼Y	0進 Y5	Y5	5進 Y5	Y5
X5	Z5	Z5	Z5	Z5
X5	Z5	Z5	Z5	Z5

2・5 進法演算の論理

4. レジスタ制御

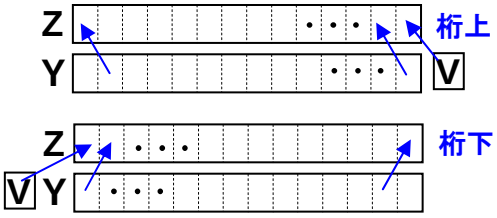
X、Y、Z の 3 本のレジスタに使われるリレーは 237 個で全リレー数の 7 割を占める。レジスタには高価なリレーを多く必要とするため、3 本レジスタだけで全ての演算を実現している。特に、乗算と除算ではレジスタ Y の分割制御などの創意工夫が施されている。

① レジスタは 3 本のみ



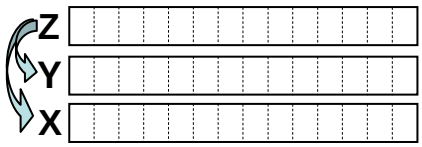
置数制御は Y と Z の 2 本、四則演算は X、Y、Z の 3 本レジスタで実現される。加減算の被演算数は X、演算数は Y に格納される。（123+456 では、X=123, Y=456）

② 桁上げと桁下げは Y の 1 桁移動⇒Z に、そして Z⇒Y で戻す



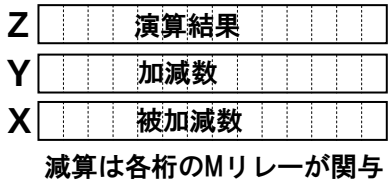
桁下げ制御は、数値キー操作でラッチ V1～V5 に蓄積された後にレジスタ Z の 14 桁目にコピーされる。続いてレジスタ Y の 14~2 桁目の数値がレジスタ Z の 13~1 桁目にコピーされて、桁下げ数値がレジスタ Z で合成された後、レジスタ Y に戻る。

③ レジスター転送（複写）は Z⇒Y と Z⇒X



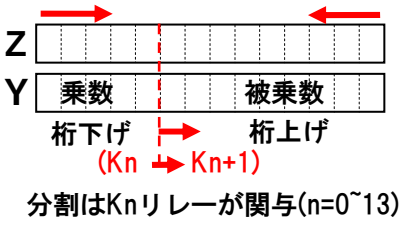
桁上桁下制御や演算結果はレジスタ Z で求まるため、レジスタ転送（複写）は Z⇒X と、Z⇒Y の 2 通りで片方向だけとなっている。

④ 演算は、X±Y⇒Z



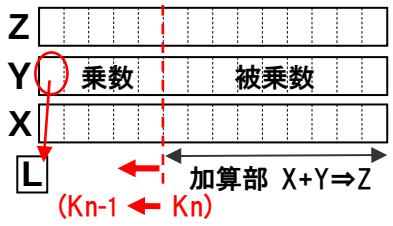
加減演算の結果はレジスタ Z に格納される。また、レジスタ X とレジスタ Y の数値は演算後も変化せず演算前の数値が残る。

⑤ 桁上げ桁下げは 2 分できる（乗算）



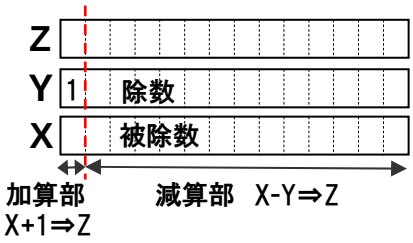
乗算では被乗数と乗数が共にレジスタ Y に格納される。乗数は 14 桁から桁下げとなるため、レジスタ Y は 2 分され、境界は置数毎に下位桁に移動する。その制御は、14 個のラッチ K0~13 から昇順移動で制御される。

⑥ 乗数 1 桁をﾀﾞｳﾝｶｳﾝﾀｰ L に転送(乗算)



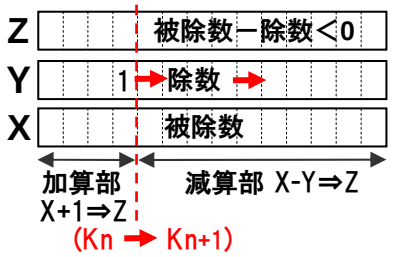
乗算では、乗数の最下位桁となるレジスタ Y の 14 桁目がラッチ L n(n=1~5)にコピーされる。ラッチ L はダウンカウンタで被乗数の加算回数を制御する。また、レジスタ Y（乗数部＋被乗数）が桁上げされる。境界も桁上するためラッチ K は降順移動となる。

⑦ 加減算は 2 分できる(除算)



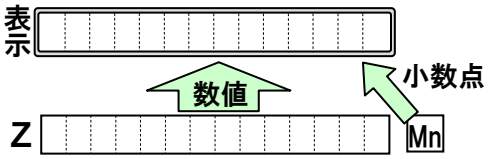
除算では、被除数と共に 12 桁目まで桁上げされる。またレジスタ Y の 14 桁目に数値 1 が代入され、減算部の減算可能回数を数える。

⑧ 減算で引けなくなると Y 桁下（除算）



除算での減算部が引けなくなったらレジスタ Y を 1 桁分桁下げする。それによって除数と数値 1 が共に桁下げとなり、境界も桁下げするためラッチ K は昇順移行となる。

⑨ Z レジスター情報が表示される

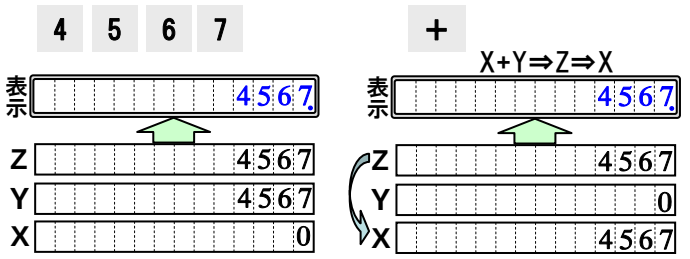


レジスタ Z の数値が表示部に導出され、各桁の該当する数値ランプが点灯する。小数点は、ラッチ Mn（n=0~13）の情報により該当桁の小数点ランプが点灯する。

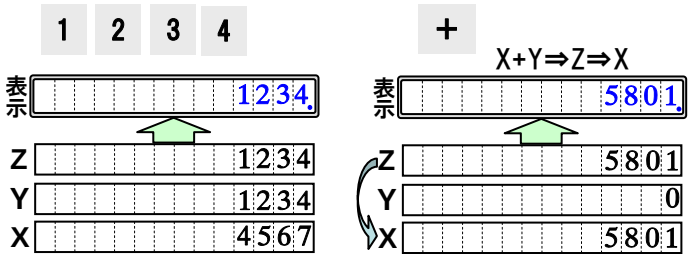
5. 加減算

加減算は乗除算を含む全ての演算の基本である。リレー計算機 14-A では、各桁毎にそれぞれ 2・5 進法の加減算回路が構成され、上位桁へキャリーと通電情報が送られる。

① 加算　例）4567＋1234＋

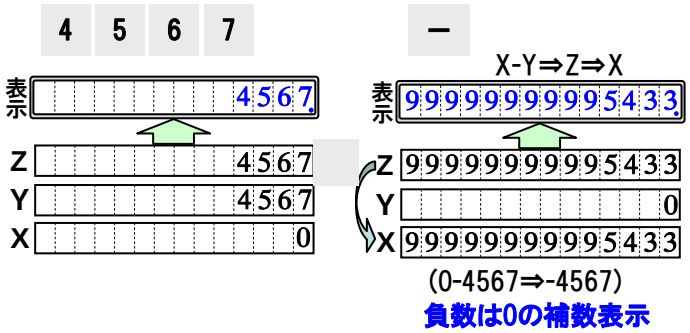


加算は＋操作で直ちに計算を実施する加算器方式で、＝操作を必要としない。電源 ON 直後では、レジスタ X は 0 のため X+Y⇒Z の加算は Y と同じとなりその値は X に戻される。

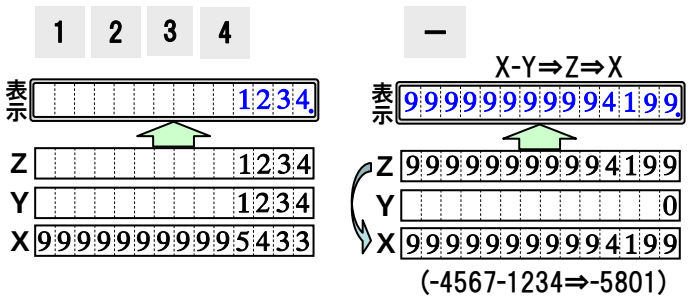


4567+1234+ の加算結果は 5801 となり、その結果は X に戻され連続加算計算が可能となる。

② 減算　例）4567－1234－

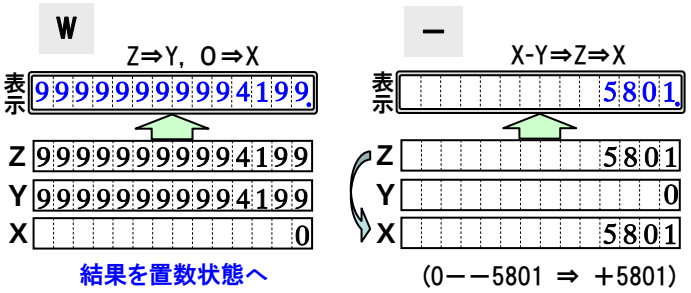


減算も－操作で直ちに計算を実施する。電源 ON 直後では、レジスタ X は 0 のため X－Y の加算は－Y となる。その結果は負数のため 0 の補数表示で表される。



4567－1234－の加算結果は－5801 で補数表示となり、その結果は X に戻され連続加減算が可能となる。

③ 負数結果を正数で見るには、

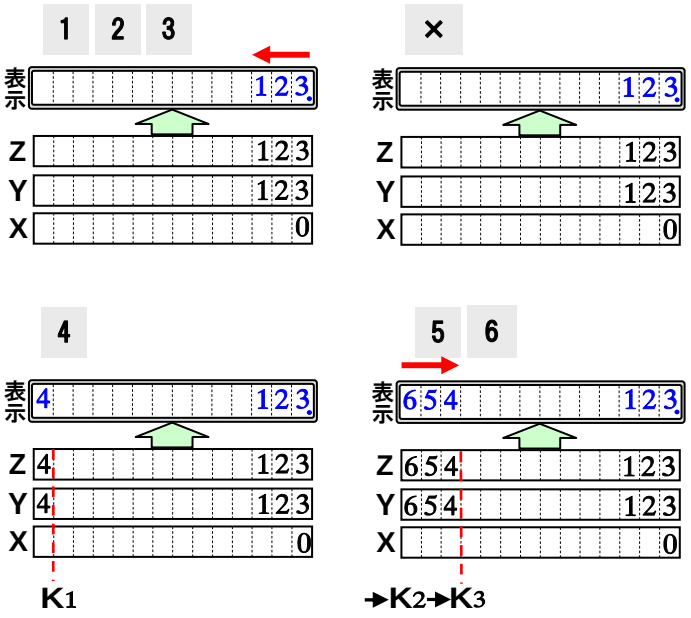


W キー操作で演算結果を置数状態に変換する。すなわちレジスタ Y に演算結果、レジスタ X には 0 が代入される。続いて－キー操作で、0－Y が計算されるため正数に戻る。

6. 乗算

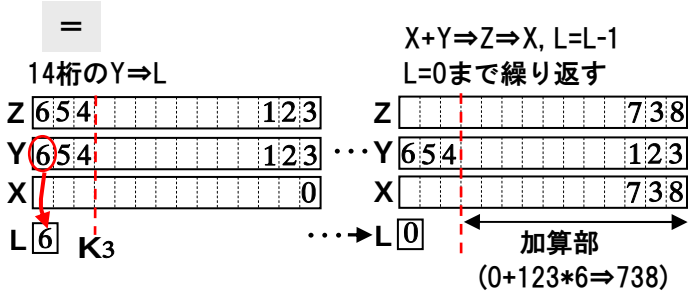
乗算は加算 X+Y の回数制御と桁上げの組み合わせで求めている。レジスタ Y は上部に乗数、下部に被乗数と 2 分される。また下部のみが加算対象で、その境界は桁上げと共に上位桁へ移動する。

① 乗算 例) 123×456 ＝

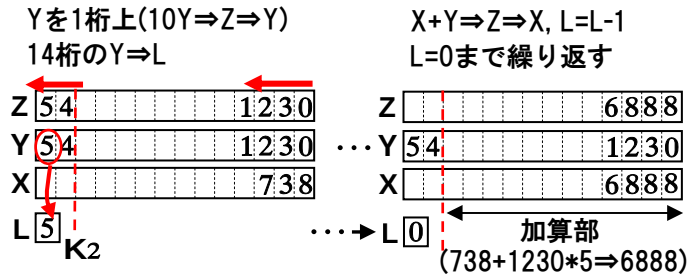


置数後の×操作で、被乗数はレジスタ Y に残り、加減算のようにレジスタ X にはコピーされない。次からの置数は乗算用の特殊置数となる。

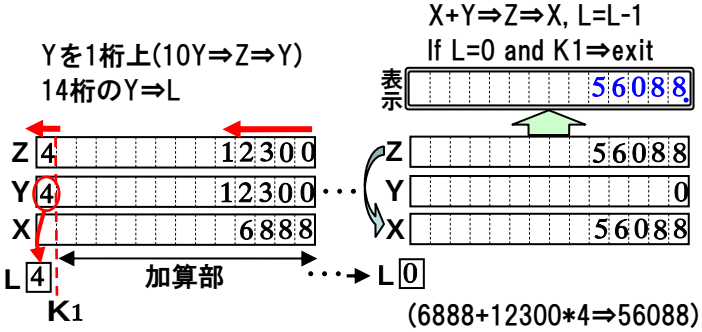
×操作後の置数はレジスタ Y の 14 桁から桁下げを伴いながら入力され、被乗数の置数とは逆となる。レジスタ Y は被乗数と乗数に 2 分され、境界は置数毎に下位桁へ移動する。ラッチ Kn（n=0~13）の昇順移行で制御される。



＝操作で乗算の計算が始まる。レジスタ Y の 14 桁目の数値が 1 桁分のラッチ Ln（n=1~5）にコピーされる。2 分された下部の加算部で X+Y⇒Z⇒X をラッチ L の数値回数だけ加算を繰り返す。L は加算毎にダウンするカウンタで 0 になったら加算を終了する。

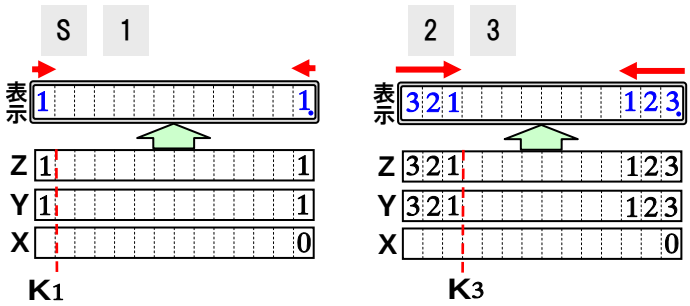


レジスタ Y 全体を桁上げにより境界も上位桁へ移動し、ラッチ Kn は降順移行する。そして同様にYの 14 桁目をLにコピーして、加算をL回繰り返す。

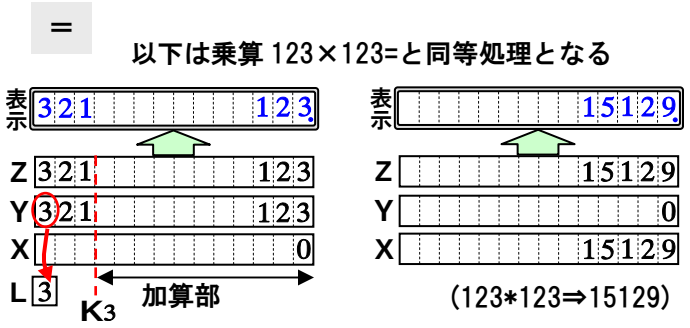


境界が K1 すなわち乗数の最上位桁まで同様な処理を続ける。最終的にレジスタ Z と X に計算結果が求まり、答えが表示される。また、乗算後はレジスタ Y は解除されゼロになる。

② 自乗計算　例) $123^2=$



自乗(Square)計算は S キー操作後の数値入力で被乗数と乗数を同時に置数処理する。最初の数値キーでレジスタ Y の 1 桁目と 14 桁目に同じ置数がなされる。続いて数値キーが操作されると被乗数は桁上げ、乗数は桁下げされた後、同様にレジスタ Y の 1 桁目と 14 桁目に同じ置数がなされる。

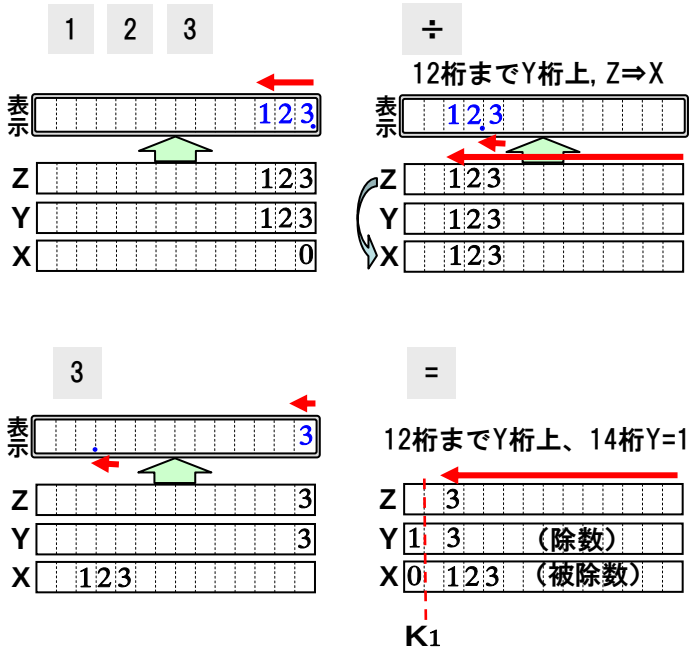


＝キー操作後は、通常の乗算と全く同じ処理となる。
左例では、123×123=15129 の計算結果が求まる。

7. 除算

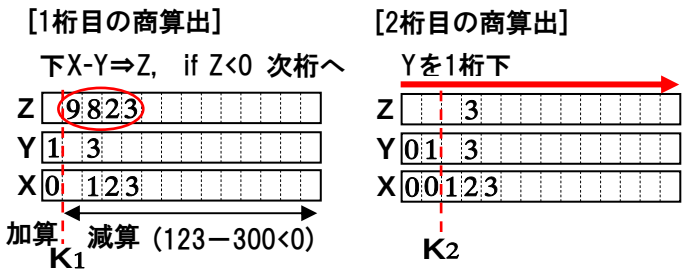
除算は減算 X−Y の引けなくなるまでの回数と桁下げの組み合わせで求めている。レジスタ X は被除数が格納され、レジスタYは上部に商、下部に除数と 2 分される。またYの下部のみが減算対象で、上部は回数計算用に 1 が代入される。その境界は桁下げと共に下位桁へ移動する。

① 除算　例 1）　123÷3 =

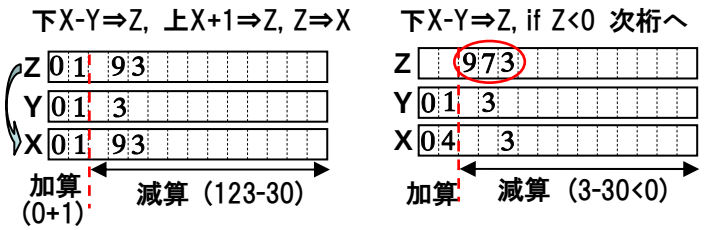


割り切れる除算を例題とする。置数後の÷操作で、置数の最上位桁が 12 桁になるまで桁上げされる。同時に小数点も桁上げされ、さらに 1 回余分に桁上げとなる。その数値は被除数としてレジスタ X にコピーされる。

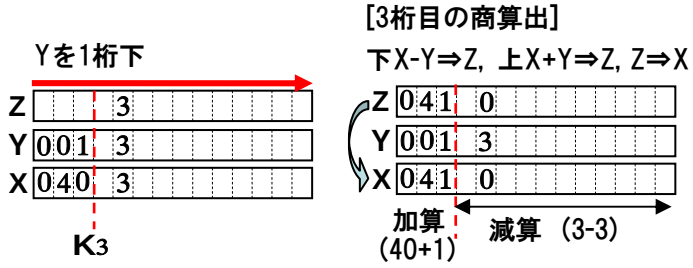
÷操作後の置数は 1 桁目からの桁上げ置数となるが、小数点も置数毎に桁上げされ、予め求める商の小数点位置を表している。＝操作で、被除数と同じく置数の最上位桁が 12 桁になるまで桁上げされ、レジスタ Y に格納される。レジスタ Y の 14 桁目に数値 1 が代入され、除数の下部と 2 分される。レジスタ X も 2 分され、下部は被除数で上部は商をカウントする。



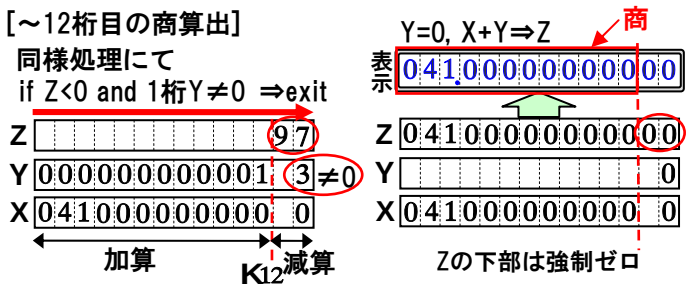
下部の X−Y が引けたら、上部の Y=1 を利用して上部の X に +1 を加える。下部の減算が引けなくなったら、次桁の商を求めるため、レジスタ Y を桁下げし、境界も桁下げ（ラッチ Kn ⇒Kn+1 に昇順移行）する。



同様に 2 桁目の商を算出する。

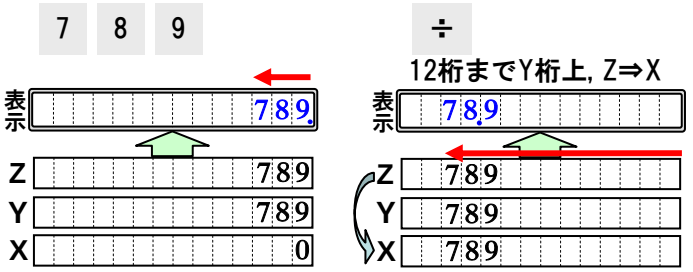


同様に３桁目の商を算出し、レジスタＹの１桁目がゼロでなくなるまで商を 1 桁ずつ算出していく。

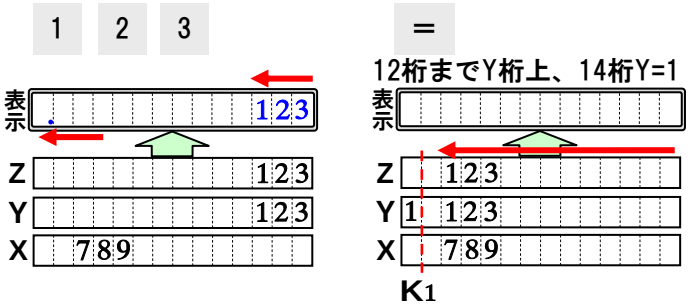


レジスタＹの１桁目がゼロ以外になったら、商の算出を終える。そしてＹは０となり、X+Y⇒Zの加算で商がＺに代入される。また２分されたレジスタＺの下部は強制的にゼロとなり、結果表示に余りの混入を防いでいる。

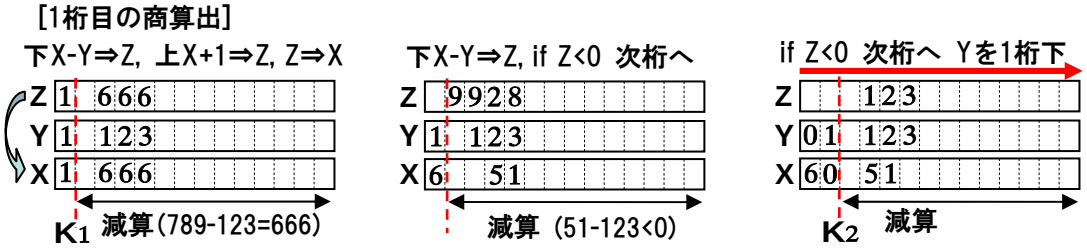
② 除算　例 2）789÷123＝

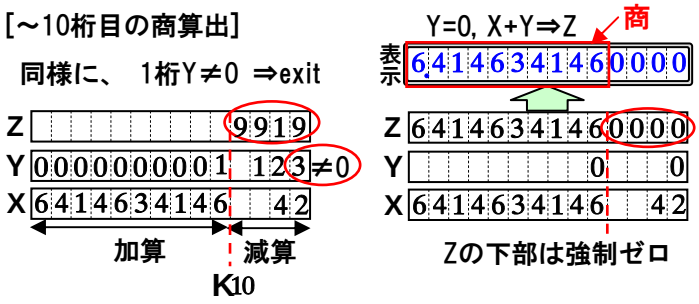


今度は割り切れない除算を例題とする。前例と同じく、÷操作と＝操作で、ともに置数の最上位桁が 12 桁になるまで桁上げされ、小数点も桁上げで商の小数点位置が求められる。



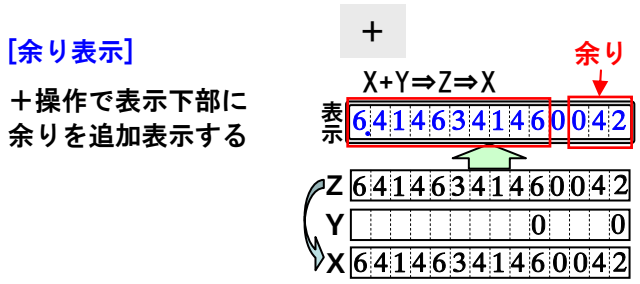
789÷123=では、レジスタＸの 14 桁目に商の最上位桁が求まる。123÷4=の場合では、レジスタＸの 14 桁目は０で 13 桁目が商の最上位桁であった。





レジスタＹの1桁目がゼロ以外になるまで商の 1 桁毎の算出を繰り返す。この例では商は6.414634146 の 10 桁が求まった。なお、レジスタ X の下部には減算の残り（余り）が格納される。

③ 余りの表示　例 3） 789÷123＝ 商・・・余り

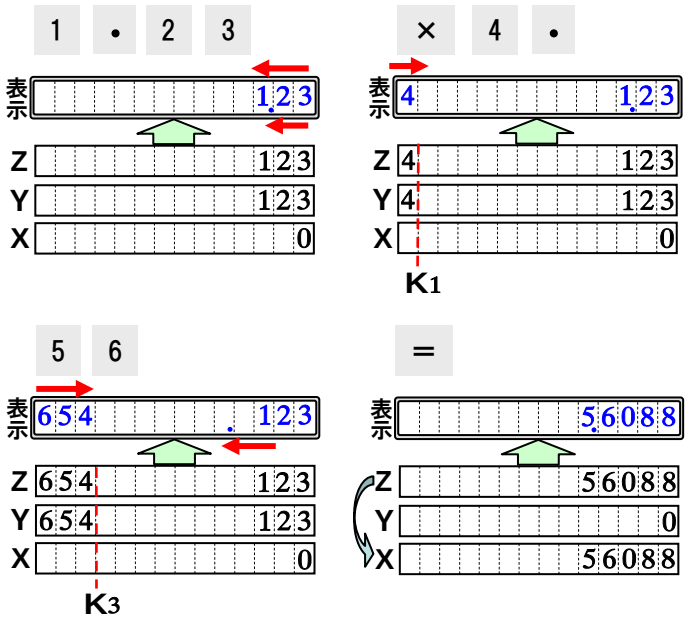


余りの表示は＋キー操作となる。割り算でのレジスタ分割状態は解除され、レジスタＹは全桁ゼロのため、X+Y⇒Z でレジスタ X の全桁情報が表示される。余りは最下位桁から除数の桁数までとなる。

8. 小数点計算

演算実施前の被演算数と演算数の置数の段階で、演算結果の小数点位置を予め求めている。また、小数点計算は乗除算に適用されている。

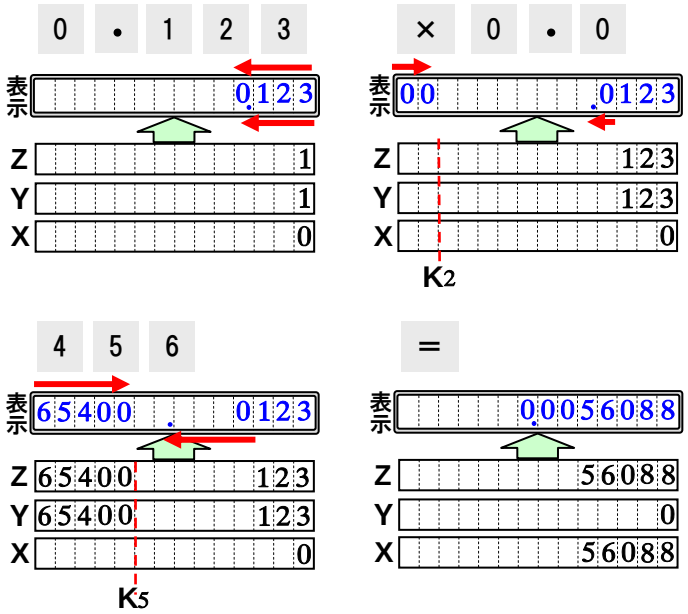
① 乗算　例 1）1.23×4.56＝



被演算置数では、小数点後の置数により、小数点位置は 1 桁ずつ桁上げされる。また乗算の演算置数でも、小数点後の置数により、小数点位置は 1 桁ずつ桁上げされる。

1.23×4.56 の計算は 0.0123×456 の計算に置換され、演算結果 5.6088 の小数点位置が予め求められたことになる。

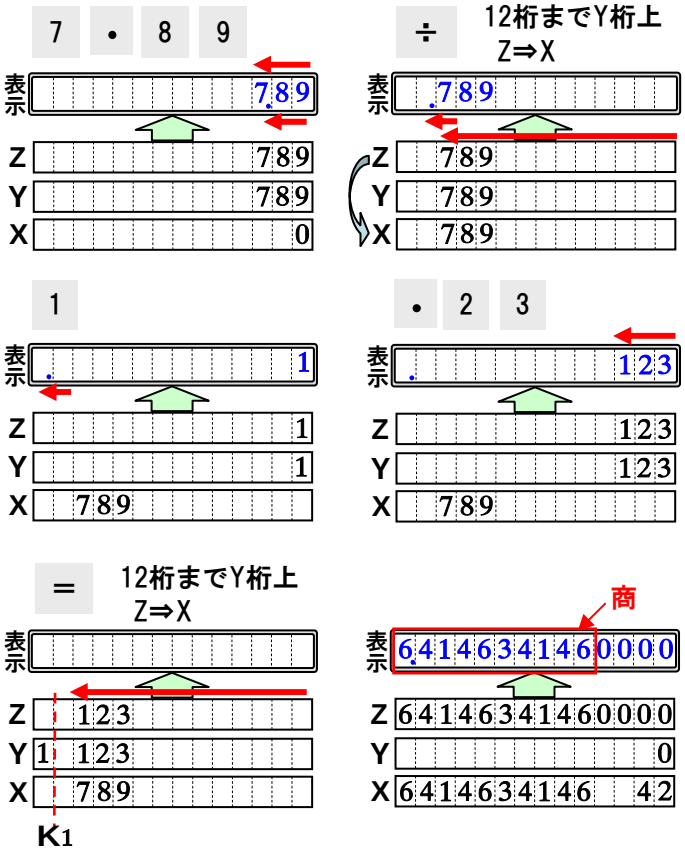
② 乗算　例 2）0.123×0.0456＝



最初の置数ゼロに続いて小数点が押される場合を追跡する。

被演算置数0.123の小数点位置は、小数点後の置数は桁上げのため 3 桁分桁上げされる。また演算置数 0.0456 の小数点位置は、小数点後の 4 つの置数にあたる 4 桁分桁上げされる。
0.123 × 0.0456 の計算は 0.0000123×456 の計算に置換され、演算結果 0.0056088 の小数点位置が予め求められたことになる。

③ 除算　例 1）7.89÷1.23＝

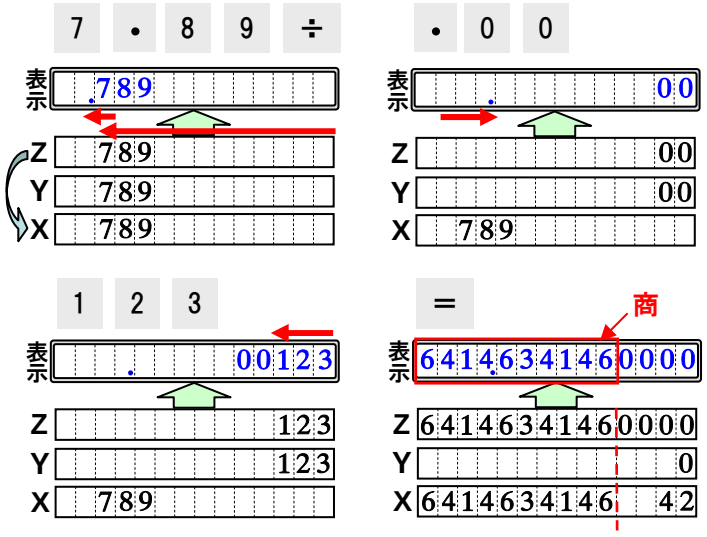


小数点を含む被演算置数の制御は、×÷などの演算操作前のため除算も乗算と同じである。

÷操作で、被演算数は 12 桁位置まで桁上げされる。小数点位置も数値の桁上分桁上げされ、さらに 1 桁分の桁上げが追加される。7.89 の場合は小数点は 13 桁位置まで桁上げされ、表示は .789 となる。

演算数の置数操作で小数点位置は桁上げとなる。小数点操作後の置数では、小数点位置は保持される。よって 1.23 の場合では、小数点位置は 1 桁分桁上げし 14 桁目となる。演算結果 7.89÷1.23=6.414634146 は最上位桁から表示され、小数点位置が予め 14 桁で求められていたことになる。

④ 除算　例 2）7.89÷.00123＝



除算の演算置数が .00123 の場合を示す。被演算置数 7.89 に続く÷操作で、小数点は 13 桁位置まで桁上げし、表示は .789 となる。
.00 のように小数点入力に続く 0 置数では、小数点位置は 0 の個数分桁下げされる。その後 0 以外の置数が操作されると、小数点位置は変化しなくなる。よって.00123 では、小数点位置は 2 桁分桁下げし、11 桁目となる。演算結果は 6414.634146 で、小数点位置 11 桁が求められたことになる。

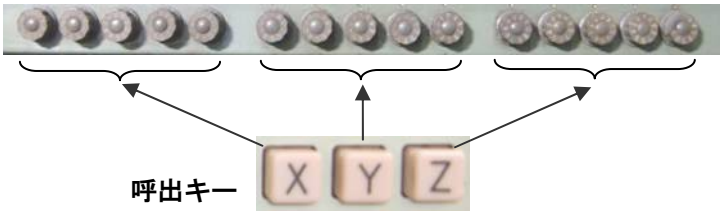
注意）除数の置数で、0.0123 と小数点入力前に 0 置数を押すと小数点位置が 1 桁ずれる。

7.89÷0.00123＝641.4634146

7.89÷ .00123＝6414.634146

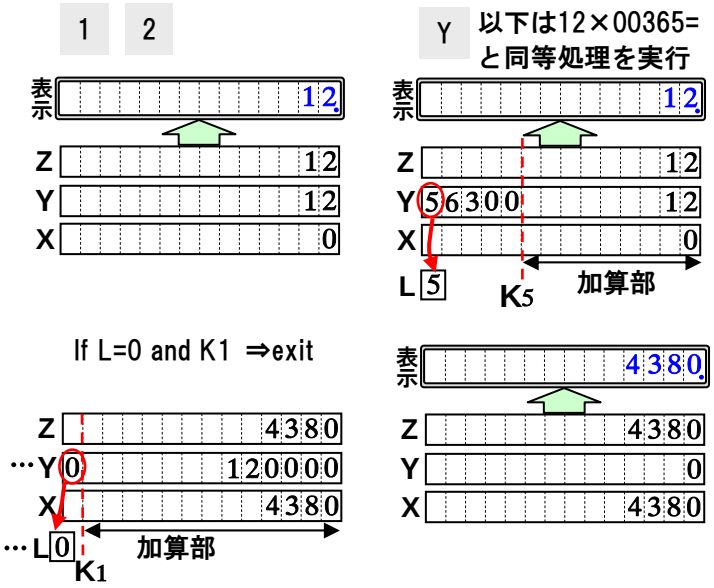
9. 定数計算

14-A には 3 種類の 5 桁数値を設定できる定数メモリーX,Y,Z が備わっている。数値の設定は手動によるダイヤル操作となる。また、この定数呼び出しは、置数後に操作パネルの右下部にある X,Y,Z キーの操作で直ちに演算を実施する。



ダイヤル

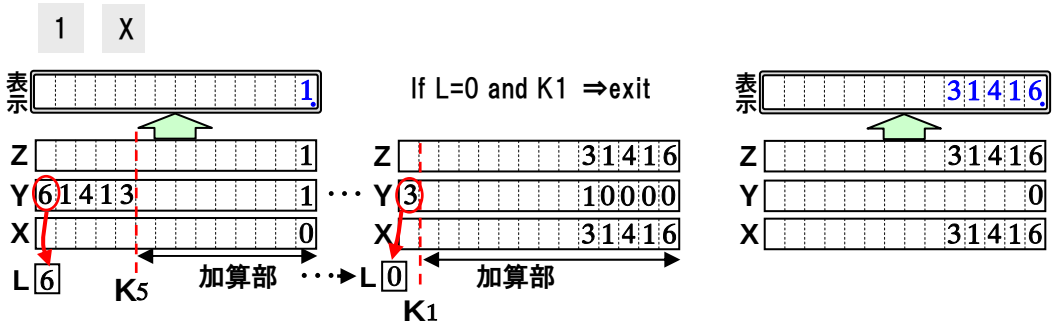
① 定数計算　例 1）12Y　（定数 Y は 00365 にダイヤル設定）



定数ダイヤル Y には 00365 が設定済みとする。置数後の Y 操作で、　置数×定数 Y＝ が直ちに実行され、キー操作を簡略化している。

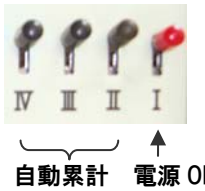
② ダイヤル設定値の確認　例 2）1X　（定数 X は 31416 にダイヤル設定）

ダイヤル設定値の確認は、数値 1 の置数後の定数操作で求まる。すなわち、1 X 操作でダイヤル X の設定値が表示される。



10. 自動累計計算

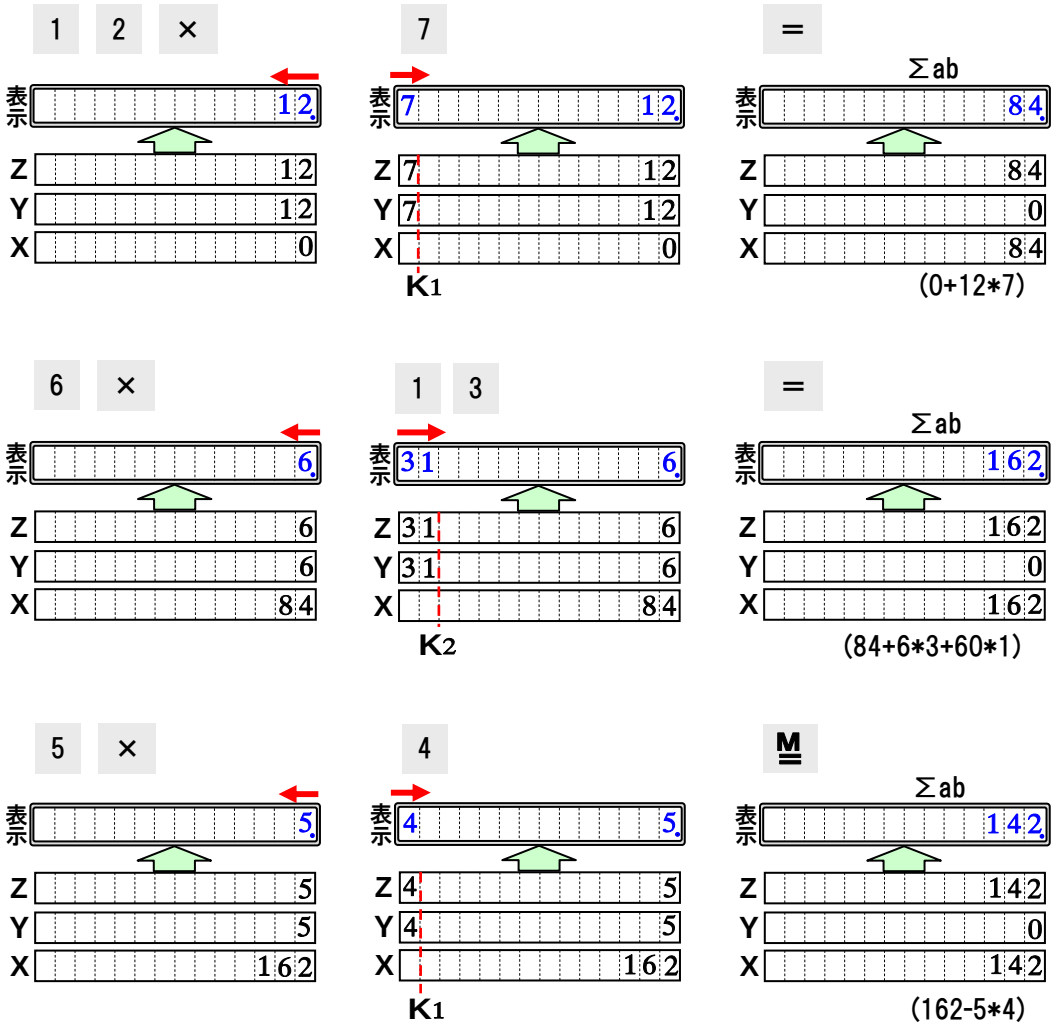
自動累計計算は、トグルスイッチでモードⅡ、Ⅲ、Ⅳのどれかを選択する。各モードの入力操作は、置数 a×置数 b＝の乗算操作の繰り返しとなる。モードⅡではΣab が表示部全桁で求まる。またモードⅢでは、表示上部7桁にΣab が、表示下部7桁に乗算 ab が求まる。さらにモードⅣでは、表示上部 5 桁にΣb が、表示下部 9 桁にΣab が求まる。



モード Ⅱ	Σab		X=X+ab
Ⅲ	Σab	ab	Xu=Xu+ab, XL=ab
Ⅳ	Σb	Σab	Xu=Xu+b, XL=XL+ab

① モードⅡ（Σab） 例1） 12×7=, 6×13=, 5×4=

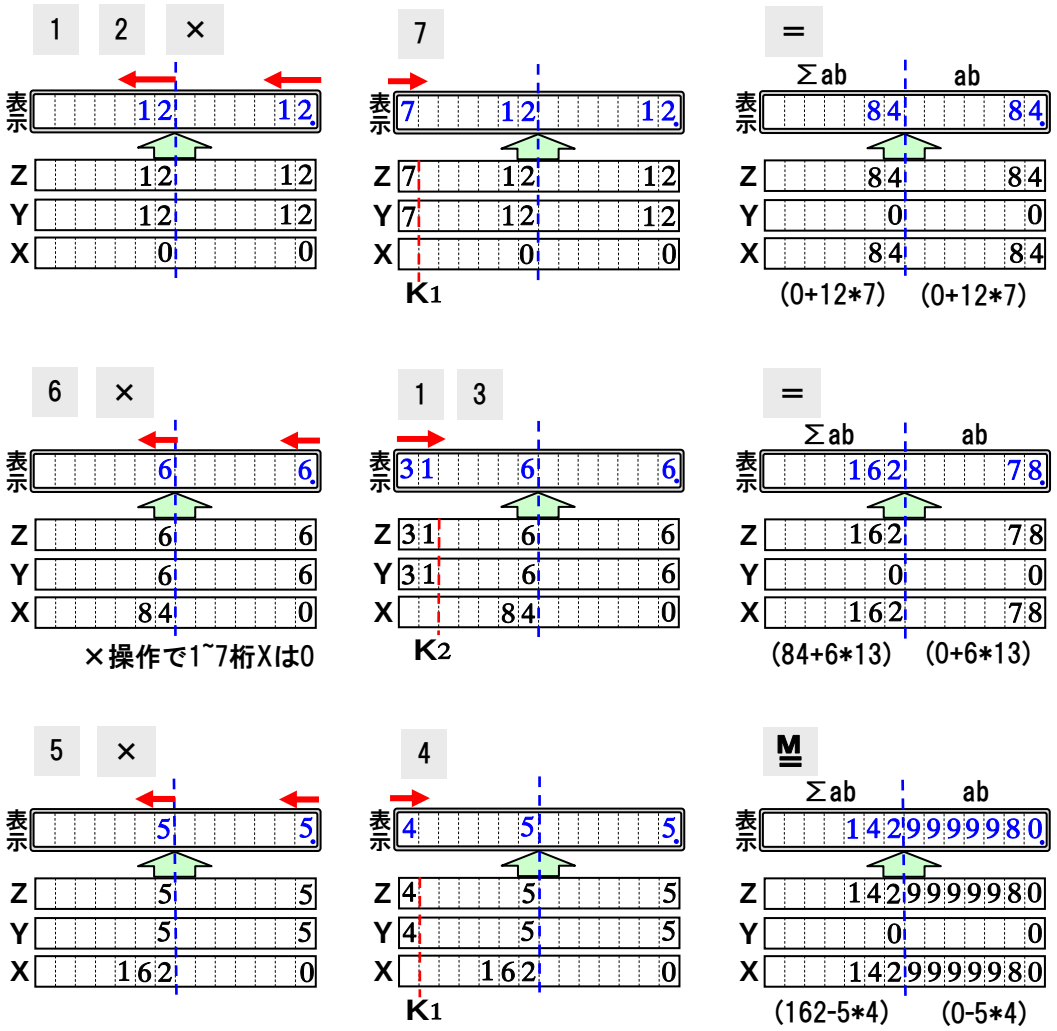
累計計算開始時は、レジスタ X はゼロ設定される。モードⅡのΣab 計算の基本は乗算と同様であるが、演算後にレジスタ X の数値が保持される。X=X+ab の計算でΣab が求まる。また減算累計の場合は、M キー操作となる。



② モードⅢ（Σab, ab） 例 1） 12×7=, 6×13=, 5×4=

モードⅢの設定により、表示中央（7 桁と 8 桁の境）に縦棒ランプが点灯し、表示上部 7 桁と下部 7 桁に均等分割される。以降、表示部及びレジスタは分割された状態で累積計算が実施される。被演算数 a の置数は、上部下部共に置数される。ただし、a の置数と共にレジスタ X の下部 7 桁はゼロに初期化される。演算数 b の置数は、通常の乗算と同じく 14 桁からの桁下げとなる。

＝操作で、レジスタ上部下部共に X＋a×b の演算が実施される。上部は Xu=Xu+ab の計算により Σab が、下部は XL=0+ab の計算により ab が求まる。演算結果の表示も分割され、負数結果の場合は、分割表示内で 0 の補数表示となる。



③ モードⅣ(Σb,Σab) 例 1) 12×7=, 6×13=, 5×4=

モードⅣの設定により、表示部の 9 桁と 10 桁の境に縦棒のランプが点灯し、表示上部 5 桁と下部 9 桁に分割される。以降、表示部及びレジスタは分割された状態で累積計算が実施される。
被演算数 a の置数は、下部のみに置数される。演算数 b の置数は、14 桁から桁下げとなるが、上部の最下桁(10 桁)に数値 1 が代入される。＝操作で、上部はXu＝Xu+1×b の計算によりΣb が、下部は、XL＝XL+ab の計算によりΣab が求まることになる。演算結果の表示も分割され、負数結果の場合は、分割表示内で 0 の補数表示となる。