Title	Transistorized digital automatic computer Keio Mark 1
Sub Title	
Author	北川, 節(Kitagawa, Misao) 都築, 東吾(Tsuzuki, Togo)
Publisher	慶応義塾大学藤原記念工学部
Publication year	1959
Jtitle	Proceedings of the Fujihara Memorial Faculty of Engineering Keio University Vol.12, No.45 (1959.) ,p.99(45)- 113(59)
JaLC DOI	
Abstract	This paper describes the systems and logical designs of the fully transistorized digital computer, named Keio Mark 1 (Abbr : K-1); the details of arithmetic and control units are described in the floating and fixed point systems of arithmetics. A high speed magnetic drum is used for the storage unit (memory). The input-output devices consist of the flexowriter, mechanical paper tape reader and punch. The connection circuits of memory and input-output devices to the computer itself are omitted.
Notes	The twentieth anniversary memorial volume
Genre	Departmental Bulletin Paper
URL	https://koara.lib.keio.ac.jp/xoonips/modules/xoonips/detail.php?koara_id=KO50001004-00120045-0045

Transistorized Digital Automatic Computer Keio Mark 1*

(Received Nov. 30, 1959)

Misao KITAGAWA**
Togo TSUZUKI***

Abstract

This paper describes the systems and logical designs of the fully transistorized digital computer, named Keio Mark 1 (Abbr : K−1); the details of arithmetic and control units are described in the floating and fixed point systems of arithmetics. A high speed magnetic drum is used for the storage unit (memory). The input-output devices consist of the flexowriter, mechanical paper tape reader and punch. The connection circuits of memory and input-output devices to the computer itself are omitted.

I. Introduction

The authors, M. Kitagawa and T. Tsuzuki started the design of this computer under the direction[1][2] of Dr. Shigeru Takahashi at the Electronics Division, Electrotechnical Laboratories, Ministry of International Trade and Industry, in July, 1958, as one of the memorial activities of the centenary of the Keio University. We completed its design in December, 1958, its constrution in May, 1959 and its adjustment in June of the same year.

The construction purpose of this computer is to do some researches concerning the programming for various calculations in the mathematical, technical and industrial fields, and the combined unit with autmatic control systems, and to train the programmers widely requested by the industrial world in Japan.

Table 1 shows the main characteristics of this computer.

---

* Original report (in Japanese) has been published in the Journal of the Institute of Electrical Communication Engineers in Japan, Vol. 42, No. 11, 1026−31, (1959)

** 北川 節, Lecturer at Faculty of Engineering, Keio University.

*** 都築東吾, Graduate student at Docter Degree Course of Electrical Engineering, Faculty of Engineering, Keio University.

(1) H. Nishino, "A Method for Designing Subunits of Serial Electronic Digital Computers", J. Inst. Elect. Commun. Engrs Japan, Vol. 38, 956−61 (1955).

(2) H. Nishino, S. Takahashi, I. Matsuzaki, K. Kondo, H. Aiso, H. Yoneda, "Transistorized Digital Computer, ETL−Mark IV", paper of Electronic Computer Tech. Committee, IECE, Japan, 1958.

Table 1. Main Characteristics of K−1

Principal System :	Serial-parallel operation, using binary coded decimal (8-4-2-1 code), Synchronized to 200 kc clock pulse, Controlled by stored program, 1− or 2−address system is adopted, Floating or fixed point arithmetic system is used.	
Word Size :	12 digits (including a sign bit); 1 digit consists of 4 bits and 1 parity check bit in parallel.	
Instructions :	1− or 2−address code, 77 distinct instructions.	
Computing Speed (not including access time) :		
	Fixed point arithmetics	
	Addition or subtraction;	0.36 ms
	Multiplication;	7.56 ms
	Division;	8.64 ms
	Comparison;	0.12 ms
	Floating point arithmetics	
	Addition or subtraction;	0.72 ms
	Multiplication;	6.36 ms
	Division;	9.24 ms
Storage Unit :	High speed magnetic drum	
	Dimension;	100 mm × 100 mm dia,
	Packing density / track;	1,200 bits
	Head for clock pulse;	2
	Head for information;	60
	Storage capacity;	1,200 words
Input-output Unit :	Consisting of :	
	Flexowriter;	1 set
	Mechanical paper tape reader;	1 set
	Mechanical paper tape punch;	1 set
Main Components :	Transistor:	900
	Germanium diode;	11,200
Power Consumption :	100 watt for computer itself.	

II. Construction of Word in K−1

II. 1. Construction of the instruction word in storage unit

As shown in Fig. 1, the instruction word consists of 12 digits: — 2 digits from the most significant digit (MSD) are the functional part where the numerical operation code is involved, and then 1 digit is used to designate the index register to be used; 4-digit address part of the next instruction involves the location number of the next instruction in the storage unit in case of 2-address system only.

In case of 1-address system, these 4 digits are not used, as they are set in zero. The address part, and in other words data address part, involving 4 gigits, holds the address number of data in the storage unit, except in such instructions as "*Shift*", "*Read*", "*Write*", "*Raise or Lower Accumulator*" etc., which are independent of the location number of storage unit. Symbol "*Δ*" at the address part of the next instruction is an indication of the 2-address system; in this case, therefore, 4,000 should be added to the location number of the next instruction. The bit marked with "×" designates a breakpoint; in this case 2,000 also must be added to the location number of the next instruction. If the break-point switch on the console is switched on, the machine will stop after the execution of that instruction. The sign digit is not used, that is, it is always positive (zero).

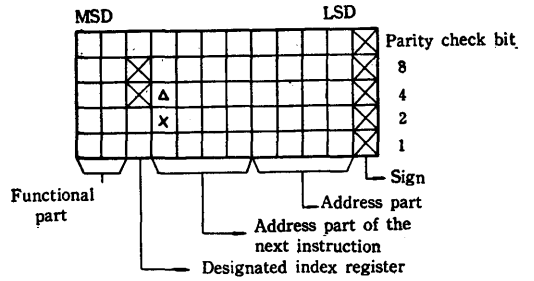

Fig. 1. Construction of the instruction word.

II. 2. Construction of the numerical word in storage unit

II. 2. 1. *Floating point representation*

As indicated in Fig. 2, numerical word z consists of sign s (1 digit), mantissa x (9 digits), and exponent y (2 digits);

$$z = (-1)^s \times x \times 10^y \qquad (1)$$

where $s=1$ when the sign is positive or $s=1$ when negative,

$1 > x \geqq 0.1$ and $99 \geqq y \geqq 0$.

In the above Eq. (1), y is a modified exponent which is given by adding fifty to the actual exponent. Thus, we can practically handle all numbers between 10^{+49} and 10^{-51} in the sense of absolute value.

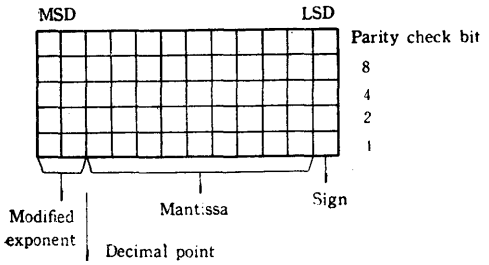

Fig. 2. Construction of the numerical word in floating point representation.

The floating point numbers represented by Eq. (1) are called the normalized or standardized form. Zero in floating point is defined as follows :

$$s = x = y = 0 \qquad (2)$$

and consequently all numbers smaller than zero defined by Eq. (2) have the sane form. A decimal point is, of course, placed at the left of the MSD of mantissa.

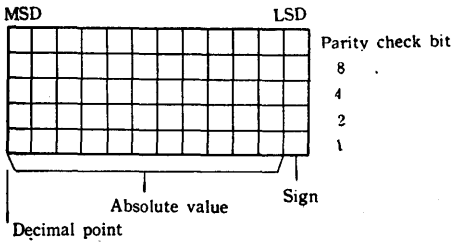

Fig. 3. Construction of the numerical word in fixed point representation.

II. 2. 2. *Fixed point representation*

The numerical words represented by fixed point consist of sign s (1 digit) and x, the absolute value (11 digits), and the decimal point is placed at the left of the MSD of the absolute value as indicated in Fig. 3. All numbers limited to $1 > |x|$ can be handled in this computer.

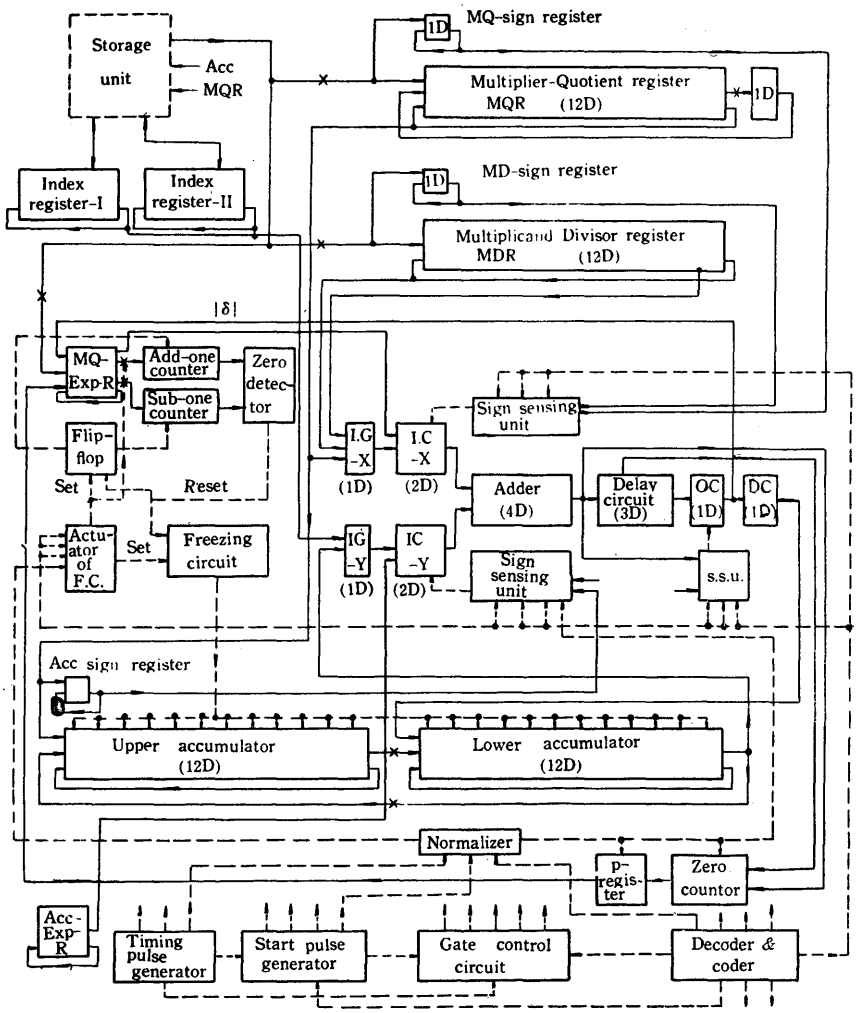

Fig. 4. Construction of the arithmetic unit.

III. Construction and Operation of Arithmetic Unit

The registers and controls used for the fixed- or floating- point arithmetic operations are shown in Fig. 4. An accumulator unit consists of upper and lower accumulator each having 12 digits respectively, thus being in a single 24-digit register in arithmetic operations or in such instructions as "*Shift*", "*Read*", "*Write*", "*Raise or Lower Accumulator*" and "*Round*". A multiplier-quotient register (MQR) has 12 digits and also 1 digits which serves to shift the contents left in multiplication or division. The multiplicand-divisor register (MDR) consising of 12 digits has not the exponent register in contrast with the other major registers. The arithmetic circuits have 12 digits in all ranging from input to output. They consist of the input gate (1 digit), input complementor (2 digits), adder (4 digits), delay circuits (3 plus 1 digits) and output complementor (1 digit). The timing pulse generator emits 12 different timing pulses which control the timings of the register components. The start pulse generator is a king of timing pulse generator which has 24 different time-phases and is used in the arithmetic operations and in such instructions as "*Shift*", "*Read*", "*Write*", "*Raise or Lower Accumulator*" and "*Round*". The gate control circuits are so provided as to control the information flow under the executions of instructions. The sign sensing units serve to determine whether the input and output complementors should be actuated or not.

Here, we describe the "*Freezing Circuit*" having a function to shift the accumulators and start pulse generator to the left. In Fig. 5, if a pulse regenerative amplifier (Abbr : PRA) "A" in each accumulator is assumed to have an output pulse simultaneously with the output pulse of the freezing circuit, the output pulse of PRA "A" is not delivered to PRA "B" through the gate "1", but to PRA "A" through its own gate "2"; thus PRA "A" is to have an output after one clock interval, that is, the information at PRA "A" is stored in "A" so long as the freezing citcuit continues to emit the output pulses. If the freezing circuit in Fig. 5 is connected to all components (or PRAs) of accumulator unit and has an output pulse at each one clock interval, the informations or contents of the accumulator unit are called to have been shifted left one digit or one position.

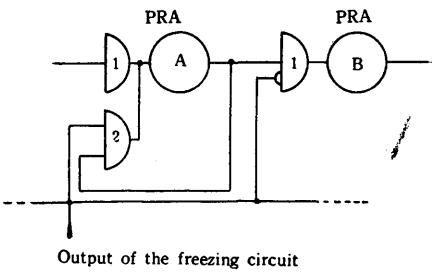

Fig. 5. Freezing circuit.

III. 1. Fixed point arithmetics

III. 1. 1. *Addition and subtraction*

An addend is brought from the designated address of the storage unit (magnetic

drum) to MQ register. The upper and lower accumulators are so placed in series as to become one 24-digit register which is called long accumulator. The addend once stored in MQ register is added algebraically to the augend stored at its upper location of long accumulator, with each digit through arithmetic circuits of 12 digits. Under operations the algebraically negative numbers are made the complements of ten or nine at the input complementor, where only the significant figure (digit) counted from the least significant digit (LSD) is made complement of ten and the other upper digits are made complements of nine. If the algebraic sum (or difference) is represented in negative by complements of ten or nine after the above operation, it is converted into the sign which is brought to the sign register, and into the absolute value which remains in the long accumulator. At the end of the execution of instruction, the long accumulator is divided into two short (12 digits) accumulators, "upper" and "lower". The information flows of this operation are shown in Fig. 6.

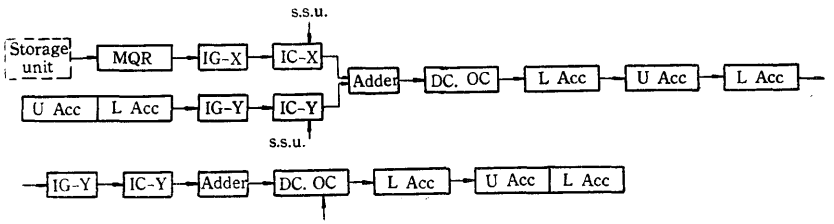

Fig. 6. Information flows in fixed point addition or subtraction.

III. 1. 2. *Multiplication*

Before this instruction is executed, a maltiplicand should be previously brought into MD register. First the multiplier itself is brought in MQ register from the designated address of the storage unit. The accumulators (upper and lower) are placed in series. By the number of times equal to the number at the MSD of MQ register, the contents of MD register is algebraically added to the contents of the long accumulator at a desired timing. The number of the MSD of MQ register is subtracted by one in each addition. Initially or after one is subtracted, the MSD of MQ register is or becomes zero respectively, when the MQ register, the long accumulator, and the start pulse generator are shifted left by one position and this operation is repeated until the contents of MQ register become zero so that MQ all zero detector is turned on. The number of shifts is stored in MQ exponent register. If the contents of MQ register are all zero, the accumulator and start pulse generator are shifted left by the number of times equal to the results of subtraction of the contents of MQ exponent register from 24.

Then the timings of the accumulator and the start pulse generator return to the initial position and hereafter the process is similar in both addition and subtraction procedures. In case of addition, subtraction or multiplication, if the contents

of accumulator overflow, the accumulator-overflow lamp (in fixed point) turns on, and the machine stops. The information flows of this operation are indicated in Block Diagram, Fig. 7.

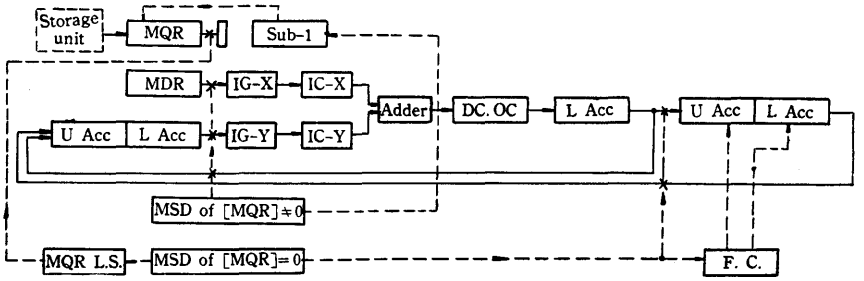

Fig. 7. Ingormation flows in fixed point multiplication.

III. 1. 3. *Division*

In this case we adopted the non-restoring method. A divisor must be previously brought into MD register. This operational sequence is as follnws : When the contents of MD register are alternatively subtracted from or added to the contents of long accumulator in an appropriate timing, one is added to or subtracted from the sign position (left of the MSD of MQ) register respectively. During the above operation, whenever the sign of the accumulator changes from positive to negative or vice versa respectively, the accumulator, MQ register and start pulse generator are shifted left by one position without losing the digit shifted out of the MSD of long accumulator. When the sign of accumulator changes 12 times and the accumulator, MQ register and start pulse generator shifted left 12 positions in all, the operation is over. The result or quotient remains in the MQ register and the remainder is set in the upper accumulator at its correct position. If the contents of MD register is equal to or larger than the contents of accumulator, a MQ-overflow lamp turns on and the machine stops. The information flows of this operation are shown in Block Diagram, Fig. 8.

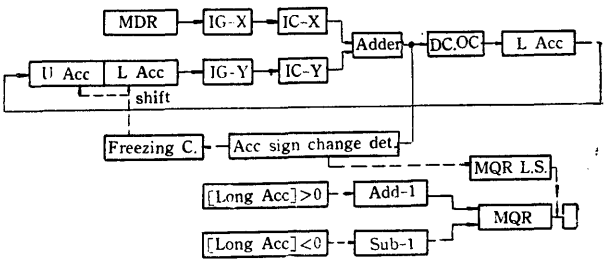

Fig. 8. Information flows in fixed point eivision.

III. 2. Floating point arithmetics

III. 2. 1. *Addition and subtraction*

The augend is in the long accumulator with the modified exponent in the exponent

register, sign in the sign register, and 18 digits or less mantissa in the long accumulator. The sign, modified exponent and mantissa of the addend are brought from the designated address of the storage unit into the MQ-sign, MQ-exponent and MQ register respectively. Then, the comparison of two exponents is, first of all, calculated through arithmetic circuits, that is,

$$(\text{Accumulator exponent})-(\text{MQ exponent})=\delta.$$

The absolute value of δ is stored in the MQ exponent register. In this connection, several combinations of the sign and absolute value of δ are considered as follows :

(1) When δ is positive and $|\delta| > 17$, the addition or subtraction is not carried out, but immediately the instruction ends.

(2) When δ is negative and $|\delta| > 17$, the contents of MQ register with its sign and modified exponent replace the contents of accumulator. In this case, the sign of accumulator is correctly converted by the combination of the MQ-sign and the instruction that means the addition or subtraction.

(3) When δ is posittve and $|\delta| \leqq 17$, at first, the long accumulator and start pulse generator are shifted left by δ positions without losing the digits shifted left out of the MSD of long accumulator and then the mantissa of MQ register is algebraically added to the contents of long accumulator. Then, the sum (or difference) is shifted left by $24-\delta$ simultaneously with start pulse generator, and moreover the long accumulator returns to its initial timing. Thereafter, the normalizing process follows the above steps.

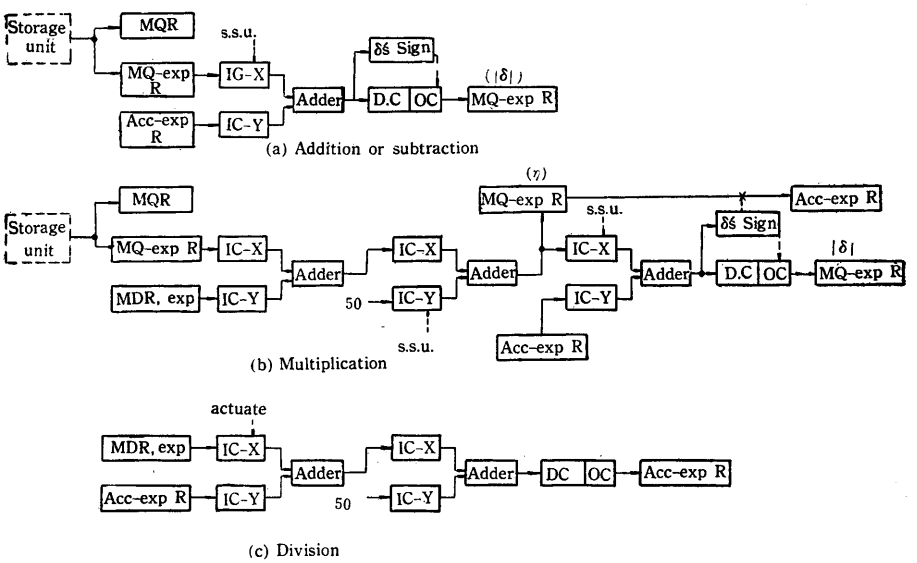

Fig. 9. Information flows in case when the exponents are calculated in floating point arithmetics.

(4) When δ is negative and $|\delta| \leqq 17$, the content of MQ exponent register replaces that of accumulator exponent register before δ is put into MQ exponent register and then the long accumulator is shifted left by $24-\delta$ positions without losing the digits shifted left out of the MSD of long accumulator.

In this case, the timing of the contents in the long accumulator will be the same as the initial timing through the gate action of the gate control unit and then immediately the normalizing process follows. The "*normalizing*" is to realign the result in the long accumulator obtained by the floating point arithmetic operations to the normalized form as indicated in Eq. (1). The procedure is as follows:

(1) If the contents of long accumulator are expressed in negative by the complements, they are converted to the sign which is brought to sign register and to the absolute value which remains in long accumulator, through arithmetic circuits.

(2) The number of zeros successively arranged from the MSD of the long accumulator is counted by the zero counter and stored in p-register, as shown in Fig. 4.

(3) The contents of accumulator exponent register are recorrected by those of p-register, and then the long accumulator is shifted left by p-positions in order to obtain normalizing form.

(4) When the mantissa overflows during the zero counting, the contents of accumulator exponent are also recorrected and the long accumulator is shifted right by one position, because the overflow of mantissa is at the most one digit on the left of the MSD of long accumulator. The unnecessary digit shifted right beyond the LSD is deleted.

III. 2. 2. *Multiplication*

A multiplicand is to be stored in MD register before the multiplication instruction is carried out. A multiplier is brought from the designated address of the storage unit into MQ register with the sign and exponent registers. The process of operation is as follows :

Since both exponents of the multiplicand and multiplier are modified by fifty, 50 should be subtracted from the sum of MD- and MQ- exponents when the exponent of the products of MD and MQ register is determined;

$$(\text{MD exponent}) + (\text{MQ expouent}) - 50 = \eta$$

η can be obtained through the arithmetic circuits, and therefore, it will replace the contents of MQ exponent register.

If η exceeds 99, the machine stops and a floating overflow lamp truns on. In respect to the operations of multiplication on mantissa, the same operations as that in the fixed point are carried out; similarly when the contents of MQ register

are detected to become all zero, the contents of the long accumulator are shifted left to its initial timing. Hereafter, the normalizing process described above is followed.

III. 2. 3. *Division*

There would coexist the exponent and mantissa in the MD register as a divisor. In this case, the quotient is in the long acuumulator without being placed in the MQ register in contrast to the case of the fixed point. The exponent calculation is preceded as follows:

$$(\text{Acc exponent}) - (\text{MD exponent}) + 50 = \xi$$

namely, the above ξ can be computed through the arithmetic circuits, and consequently it will replace the contents of the accumulator exponent register.

In this connection, care must be taken that 50 should be added to the difference between the accumulator and MD exponents. The dividing process of mantissa is the same as in the fixed point division, but the quotient once placed in the MQ register is transferred to the long accumulator and then the normalizing procedure is followed; the remainder should be neglected and eliminated in the accumulator.

As to the information flows in floating point arithmetics, especially the calculations of exponent are shown in Fig. 9.

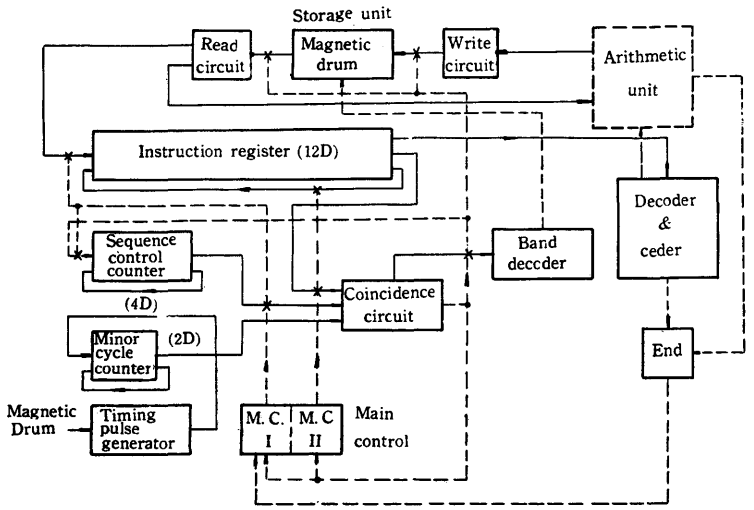

Fig. 10 Construction of the control unit

IV. Construction and Operation of Control Unit

The block diagram of control unit is shown in Fig. 10. A minor cycle counter (MCC) is a counter with the module of 100.

The MCC contents can be increased by one at every one word-time, thus deciding the rotated position of the magnetic drum in each revolution. A 4-digit sequence control counter (SCC), having the address number of the storage unit storing the next instruction to be executed, is usually increased by one at each time of transferring one instruction from the storage unit to the instruction register. The computation is executed successively in stages, I and II alternatively; in the stage I one instruction is transferred from the storage unit to the instruction register, and in the stage II the instruction stored in the instruction register is executed.

Since the magnetic drum consists of twelve bands, each one containing five tracks and 100 locations, the contents of MCC are always referred to the address number of the location in every band. In the stage I, the contents of the location addressed by the SCC contents are read out to the instruction register under such heads of the magnetic drum as are actuated by the band selector circuits through the band decoders referring to the higher 2 digits of SCC, at the time of coinciding the lower 2 digits of SCC with the MCC contents through a coincidence circitits. The coincidence pulse emitted in the above process increases the SCC contents by one and is also delivered to the main control unit for changing the stage from I to II.

In the stage II, the functional part of one instruction is transferred to the decoder and also its address part to the coincidence cricuits and band decoder. If the instruction has any relation to the magnetic drum, the operations similar to the stage I takes place to read the information out of the storage unit by using the address part of the instruction instead of the SCC contents mentioned above. In case of such instructions as having no relation to the storage unit, they are immediately executed after decoding. When an index register is designated according to the instruction, the address part will be modified by the contents of the designated index register before the instruction execution. When an instruction has been executed in any way, an end pulse is emitted and the stage is changed from II to I, and the contents of the instruction register are cleared.

The K—1 computer can execute 77 distinct instructions tabulated in Table 2. When the start key is pressed, the contents of sequence control counter are cleared and the stage is set to I; then the initial program addrassed in 0 to 2 in the magnetic drum can read the instructions and the data punched on a paper tape and store them in the programmed locations. Then the execution of the program begins.

V. Conclusion

Now we have described the system and logical designs of K—1 computer in fixed and floating arithmetics and also the instructions executed by this computer in some details. We omitted the explanations concerning the input-output devices and storage unit. The practices in utilizing and manipulation of this computer

are also left to be explained in its instruction manual which will be published in the near future. We did our best in constructing this computer within the limits of our abilities, but it will be reinforced by a large capacity core menory and high speed input output devices to make possible better utilization. However, even now it has several interesting features as mentioned below.

(1) It can handle fixed and floating point arithmetics mixed in one program.
(2) It can do the same with 1—address and 2—address code systems.
(3) Simple "initial program" is used for reading the progammed paper tape.

Acknowledgement

We sincerely express our gratitudes to Dr. Fumiki Kito, the Dean of the Faculty of Engineering, Prof. Tomoyuki Somiya and Dr. Shigeru Takahashi for giving us the opportunity to construct this computer.

We also wish to acknowledge our appreciations to all the members of this Faculty who encouraged us with warm advices in public and in private. Thanks are expressed particularly to Mrs. Yasuyo Arai, Tetsuo Hara, Tatsuo Kudo, Takuo Shinkai and Kiyotaka Sakai, graduates of this Faculty in 1959, who assisted us in constructing this computer.

Table 2. Instructions of K−1

Note :
- I : Designated index register (1 or 2).
- A : Designated location numcer addressed in the storage unit.
- B : Number placed in the address part of an instruction for special use.
- E : Effective address, which means A plus the contents of the designated index register I.
- Acc : Accmulator
- MQR : Multiplier-quotient register
- MDR : Multiplicand-divisor register
- SCC : Sequence control counter
- [x] : The contents of x.

Operation Code	Operation
01 I A	The machine stops. When the re-start button on the console is pressed the next instrution is taken from location E.
02 I A	Next instruction is taken from lacation E, regardless of where the present instruction is.
03 I A	If [Acc] >0, next instruction is taken from location E; if not, next instruction is taken in normal sequence.
04 I A	If [Acc] <0, next instruction is taken from location E; if not, next instruction is taken in normal sequence.

05 I A If [Acc] = 0, next instruction is taken from location E; if not, next instruction is taken in normal sequence.

06 I A If [Acc] ≠ 0, regardless of Acc sign, next instruction is taken from location E; if not, next instruction is taken in normal sequence.

10 Quotient, [long Acc] / [MDR] in fixed point is placed in MQR; the remainder is in upper Acc. If MQR overflows, the machine stops.

11 Quotient, [long Acc] / [MDR] in floating point is placed in long Acc; the remainder is lost. If Acc overflows, the machine stops.

12 Round long Acc at the MSD of lower Acc in fixed point and lower Acc is cleared.

13 Round [long Acc] at the MSD of lower Acc in floating point and lower Acc is cleared.

14 B Acc exponent is increased by B.

15 B Acc exponent is decreased by B.

16 [upper Acc] replaces [MDR] in fixed point; upper Acc is unchanged.

17 The contents of upper Acc with its exponent replace [MDR] in floating point; upper Acc and exponent are unchanged.

20 Logical products of every digit of upper Acc and the corresponding digit having one in MDR remain in upper Acc; MDR is unchanged.

21 The sign of Acc is changed to its opposite sign.

22 By use of mechanical paper tape reader 1, every one word which consists of 12 characters in maximum up to the termination code on a paper tape is read into the input register for each character, and first four characters (sign and 3 higher significant digits) are once read in MQR and the rest 8 (maximum) characters are read in upper Acc. Hereafter this instruction makes several different behaviors according to the functions of the termination codes α, β, γ, δ, π and θ, as explained below :

α : When this termination code is read in input register, 4 characters once read in MQR are transferred to the corresponding positions of upper Acc to compose them into one word with the contents of upper Acc and the address part of upper Acc replaces the contents of index registers 1 and 2; then SCC is cleared.

β : This termination code is used only in 2-address code system. 4 characters once read in MQR are transferred to the corresponding positions of upper Acc to compose them into one word with the contents of upper Acc. The contents of index register 1 replace those of index register 2, and 4 characters corresponding to the next instruction address replaces the contents of index register 1.

γ : 4 charadters read in MQR are left as they are. The address part of [upper Acc] replaces [SCC].

δ : 4 characters read in MQR are left as they are. The address part of [upper Acc] replaces [SCC] and the machine stops.

π : 4 characters in MQR are transferred to the corresponding positions of upper Acc to compose them into one word with the contents of upper Acc. The address part of [upper Acc] is modified by the contents of index register 1.

θ : Same as the operation in π except that the address part of [upper Acc] is not modified by the contents of index register 1.

23			Same as operations in code 22 except by using the mechanical paper tape reader 2.
24		B	By using the mechanical tape reader 1, characters counted by B on the paper tape are read in the long Acc from the LSD of upper Acc.
25		B	Same as the operation of code 24 except by using the mechanical tape reader 2
26		B	Characters counted by B from the MSD of long Acc are delivered to the output devices.
27		B	The character specified by B is delivered to the output devices.
28		B	[long Acc] are shifted left B places. Digits out of the MSD of upper Acc are lost. Zeros are entered at the LSD of lower Acc as the number is shifted.
29		B	[long Acc] are shifted right B places. Digits out of the LSD of lower Acc are lost. Zeros are entered at the MSD of upper Acc as the number is shifted.
40	I	A	[E] is algebraically added to [long Acc] at its upper place.
41	I	A	[E] replaces [upper Acc], and lower Acc is cleared.
42	I	A	[E] is algebraically subtracted from [long Acc] at its upper place.
43	I	A	−[E] replaces [upper Acc], and lower Acc is cleared.
44	I	A	The algebraic product [MDR] × [E] is added algebraically to [long Acc].
45	I	A	The algebraic product [MDR] × [E] replaces [long Acc].
46	I	A	The algebraic product [MDR] × [E] is subtracted algebraically from [long Acc].
47	I	A	The negative of the algebraic product [MDR] × [E] replaces [long Acc].
48	I	A	[upper Acc] replaces [E], [upper Acc] is unchanged.
49	I	A	Zero replaces [E].
50	I	A	In floating point, [E] is algebraically added to [long Acc].
51	I	A	In floating point, [E] replaces [upper Acc]; lower Acc is cleared.
52	I	A	In floating point, [E] is algebraically subtracted from [long Acc].
53	I	A	In floating point, −[E] replaces [upper Acc], lower Acc is cleared.
54	I	A	In floating point, the algebraic product [MDR] × [E] is algebraically added to [long Acc].
55	I	A	In floating point, the algebraic product [MDR] × [E] replaces [long Acc].
56	I	A	In floating point, the algebraic product [MDR] × [E] is algebraically subtracted from [long Acc].
57	I	A	In floating point, the negative of the algebraic product [MDR] × [E] replaces [long Acc].
58	I	A	[Upper Acc] with its exponent expressed by floating point replaces [E].
59	I	A	Zero replaces [E].
60	I	A	[E] is algebraically added to [long Acc] and the quotient, [long Acc]/[MDR] is obtained in MQR; the remainder is in upper Acc.
61	I	A	[E] replaces [long Acc] and the quotient, [long Acc]/[MDR] is obtained in MQR; the remainder is in upper Acc.
62	I	A	[E] is algebraically subtracted from [long Acc] and the quotient, [long Acc]/[MDR] is obtained in MQR; the remainder is in upper Acc.
63	I	A	−[E] replaces [long Acc] and the quotient, [long Acc]/[MDR] is obtained in MQR; the remainder is in upper Acc.
64		B	Number B is added to [long Acc] at the address part of upper Acc.
65		B	Number B replaces [long Acc] at the address part of upper Acc.

66		B	Number B is subtracted from [long Acc] at the address part of upper Acc.
67		B	$-B$ replaces [long Acc] at the address part of upper Acc.
68	I	A	The address part of [upper Acc] replaces the corresponding position of [E]; upper Acc and the other positions of E are unchanged.
69	I	A	Zero replaces the address part of [E]; the other positions of [E] are unchanged.
70	I	A	In floating point, [E] is added algebraically to [long Acc] and the quotient, [long Acc]/[MDR] is obtained in long Acc; the remainder is lost.
71	I	A	In floating point, the quotient, [E]/[MDR] is obtained in long Acc; the remainder is lost.
72	I	A	In floating point, [E] is subtracted algebraically from [long Acc] and the quotient, [long Acc]/[MDR] is obtained in long Acc: the remainder is lost.
73	I	A	In floating point, the quotient, $-$[E]/[MDR] is obtained in long Acc; the remainder is lost.
74	I	A	[MQR] replaces [E]; [MQR] is unchanged.
76	I	A	[E] replaces [MDR]; [E] is unchanged.
78	I	A	The address part of [E] replaces the corresponding positions of [upper Acc]; other positions of [upper Acc] are unchanged.
79	I	A	[Acc] becomes zero and the address part of [E] replaces the crresponding positions of [upper Acc].
80	I	A	If [I] $\neq 0$, next instruction is taken from location A; if not, next instruction is taken in normal sequence.
81	I	A	If [I] $= 0$, next instruction is taken from location A; if not, next instruc- is taken in normal sequence.
82	I	A	If [I] > 0, next instruction is taken from location A; if not, next instruction is taken in normal sequence.
83	I	A	If [I] < 0, next instruction is taken from location A; if not, next instruction is taken in normal sequence.
84		A	If the contents of index register 1 do not equal to those of index register 2, next instruction is taken from location A; if two index registers have the same contents, next instruction is taken in normal sequence.
85	I	A	[SCC] replaces [I] and next instruction is taken from locotion A.
86	I	A	The address part of [A] replaces [I].
87	I	A	[I] replaces the address part of [A] and the other part of [A] is unchanged.
88	I	B	B is added to [I].
89	I	B	B is subtracted from [I].
90	I		[I] replaces the address part of [upper Acc]; the other part of [upper Acc] is unchanged.
91			[MQR] replacees [upper Acc].
92	I	B	Number B replaces [I].
93			If each digit of [upper Acc] equals to the corresponding digit of [MDR], the equal digits only remain in upper Acc in their initial positions; unequal digits are lost in upper Acc; [MDR] is unchanged.
98	I		The address part of [upper Acc] replaces [I] regardless of the sign of Acc.