UDC 621.316.7 : 681.325

FUNCTIONAL EXPANSION SYSTEM OF FUJI SEQUENCER

Masahiro Yoshida
Keiichi Tomizawa
Ken Hagiwara

I. INTRODUCTION

The sequencer is basically a dedicated sequence controller having logical processing functions. However, the sequencers are frequently required to have arithmetic functions and a means of data transmission to extend their applications, such as a composite system of sequence control and simple DDC, a distributed control system of sequencers in large or wide-range systems, or a hierarchy control system with computer.

One of the Fuji sequencers (SC-20) has been designed with a functional building block system and permits addition of the above functions by adding the function modules optionaly.

The method of expanding the hardware functions and the outline of the software system are given in this paper.

II. FUNCTIONAL EXPANSION METHOD

The sequencer's functions can be expanded as shown in *Fig. 1* by adding function modules. Since each function module has a common interface and the same external dimensions with general I/O PC cards, they can be inserted at an optional I/O card position (address).

Data exchange between the central processing unit (CP) and the function modules can be performed by means of the program mode and DMA mode.

The control flags and the data of the function module can be managed by the program in the same manner as the general I/O modules (program mode). Moreover, the function module can have direct access to the core memory and transfer to or from the core memory without regard to the sequencer program. The CP of the sequencer interrupts the execution of the program by receiving the DMA request from the function module, and then the function module performs direct data exchange with the core memory.

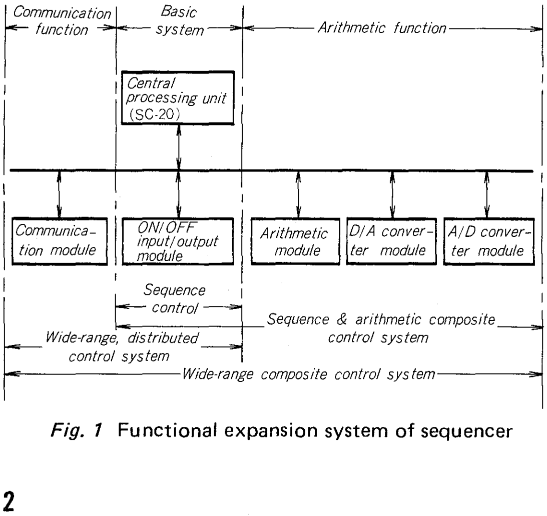

Communication function
Basic system
Arithmetic function
Central processing unit (SC-20)
Communica-tion module
ON/OFF input/output module
Arithmetic module
D/A conver-ter module
A/D conver-ter module
Sequence control
Sequence & arithmetic composite control system
Wide-range, distributed control system
Wide-range composite control system

Fig. 1 Functional expansion system of sequencer

Direct data exchange between function modules is impossible, but it is possible to exchange the data via the data area of the core memory. Also, output or input of external signals related to the function module is possible via the core memory.

The function modules include the arithmetic module, A/D module, D/A module and communication module.

III. ARITHMETIC MODULE

The arithmetic module expands the functions of the sequencer up to the arithmetic control range.

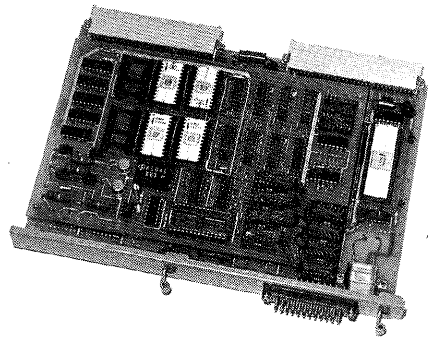


Fig. 2 (a) Arithmetic module

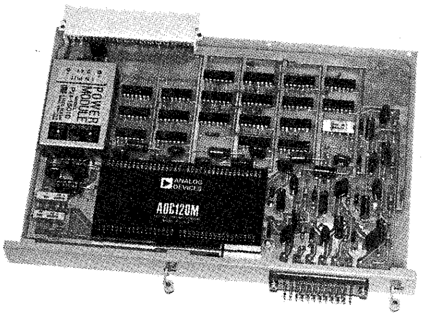


Fig. 2 (b) A/D converter module

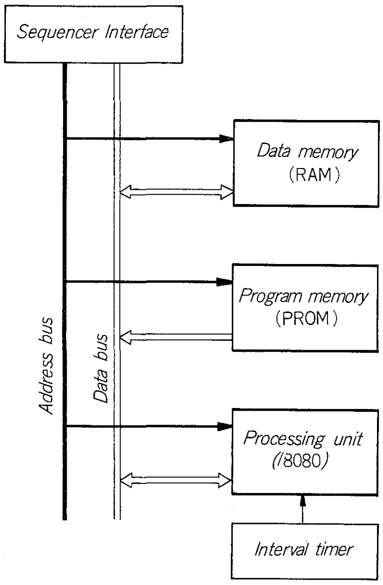


Fig. 3 Block diagram of arithmetic module

A sequencer incorporating this module is composed of two processors–one for the intrinsic sequence control and the other one for arithmetic control. Data exchange between the two processors can be performed by means the program of the sequencer or DMA mode of the arithmetic module.

The airthmetic module is compactly designed to be accommodated in the space of one I/O card, An external view is shown in *Fig. 2,* and the functional structure is shown in *Fig. 3.* It consists of a program memory (EP-ROM), data memory (RAM), processing unit (PU) and sequencer interface. An Intel 8080 microprocessor is used as the processing unit.

IV. A/D, D/A MODULE

These modules add an analog interface to the input-output of the sequencer and operate mainly with the arithmetic module.

The specifications of the A/D module and D/A module are given in *Table 1* and *Table 2* respectively.

The A/D module has 8 channel analog input and converts 0~4.995V to BCD 3 digit (0~999). The D/A module has 8 channel analog output in the case of converting BCD 3 digit (0~999) to 0~4.995V and in the case of converting BCD 3 digit (–999~0~999) to –4.995~0~4.995V, limited to 4 channel outputs.

V. COMMUNICATION MODULES

Data transmission between sequencers, sequencer and computer and sequencer and data highway can be performed by adding the communication module to the sequencer.

The communication module consists of one transmit module and one receive module. The transmit module is located on the transmission side and the receive module on the receiving side.

The specifications are listed in *Table 4* and the features are summarized below:

Table 1 Specifications of A/D converter module

Item	Specifications
Number of input channels	8 channels/module
Input signal	0~ 5V, 0V is grounded inside the module
Input resistance	10 MΩ or greater
Time constant of input filter	100 ms
Conversion characteristics	0~4.995V→BCD 3 digits, 0~ 999
Conversion time	Within 40μs
Imput connection	Multi connector (34 pin)
Power supply	Sequencer I/O power supply
Dimensions	Conforms to sequencer I/O card

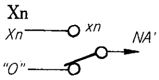
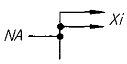
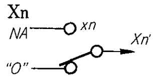
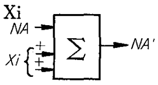
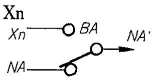
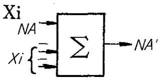
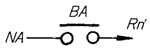
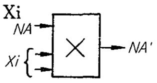
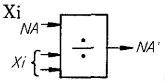
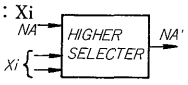
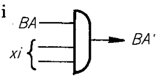
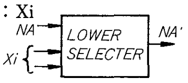
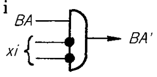
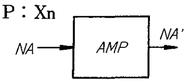
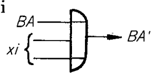
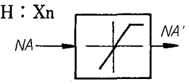
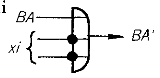
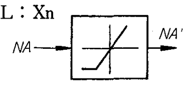
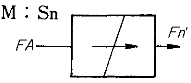
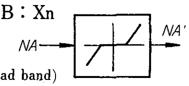
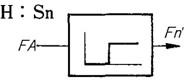
Table 2 Specifications of D/A converter module

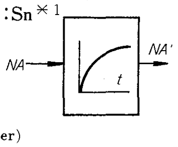
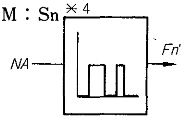
Item	Specifications
Number of output channels	8 channel (single polarity) or 4 channel (double polarity)
Output signal	0 ~4.995 V, 1mA or less
Time constant of output filter	100 ms
Conversion characteristics	BCD 3 digits 0~999→0~4.995V or –999~0~999→ –4.995~0~4.995
Output connection	Multi connector (34pin)
Power supply	Sequencer I/O power supply
Dimensions	Conforms to sequencer I/O card

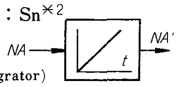
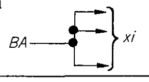
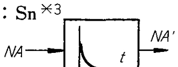
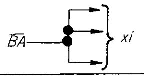
Table 3 Specifications of communication module

Item	Specifications
Transmission system	Word serial, block transmision (max. 256W) asynchronous system
Transmission data	SC-20 core memory (I/O buffer, timer, counter and program area)
Check system	Parity check, repetitive transmission collation
Transmission speed	Max. 5 kW/s
Transmission distance	Max. 1km
Input signal	Photo-coupler isolated no voltage ON/OFF signal dc 24V 10mA
Output signal	Isolated transister open collector signal dc 24V 100mA
Insulation lever	ac 500V, 1 minute
Power supply	dc 24V (SC I/O power supply)
Dimmensions	Conforms to sequencer I/O card
Connector	Multi connector (60pin)

Table 4 Macro-instructions

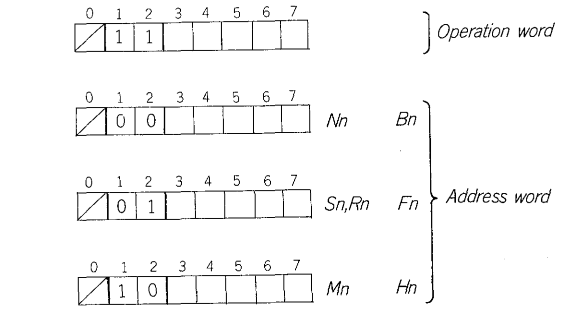
No.	Symbol	Operation	No.	Symbol	Operation
0	INA : Xn Xn⟶ NA	Xn→NA'	16	SWI : Xn 	When xn= 1, Xn→NA' When xn= 0, O→NA'
1	ONA : Xi 	NA→Xi	17	SWO : Xn 	When xn= 1, NA→Xn' When xn= 0, O→Xn'
2	SUM : Xi 	NA+ΣXi→NA'	18	SWX : Xn 	When BA= 1, Xn→NA' When BA= 0, NA→NA'
3	SUB : Xi 	NA−ΣXi→NA'	19	SET 	When BA= 1, NA→Rn' (Initialize Rn)
4	MLT : Xi 	NA× $\left(\frac{Xi}{1000}\right)$ →NA'	20	RP : xn	xn→BA
5	DIV : Xi 	NA÷ $\left(\frac{Xi}{1000}\right)$ →NA'	21	RN : xn	$\overline{xn}$→BA
6	HS : Xi 	Max. { NA, Xi } →NA'	22	AP : xi 	BA ∧ xi→BA
7	LS : Xi 	Min. { NA, Xi } →NA'	23	AN : x i 	BA ∧ $\overline{xi}$→BA
8	AMP : Xn 	NA×Xn→NA'	24	OP : xi 	BA ∨ xi→BA
9	SAH : Xn 	When NA≦Xn, NA→NA' When NA<Xn, Xn→NA'	25	ON : xi 	BA ∧ xi→BA
10	SAL : Xn 	When NA≧Xn, NA→NA' When NA<Xn, Xn→NA'	26	TIM : Sn 	When BA= 0, 0→Rn', Fn' When BA= 1at Fn=0,Rn+1→Rn' if Rn≧Sn, 1→Fn
11	DEB : Xn  (Dead band)	When NA−Xn> 0, NA−Xn→NA' When NA+Xn< 0, NA+Xn→NA' When−Xn≦NA≦Xn, 0→NA'	27	CPH : Sn 	When NA≧Sn 1→Fn' When NA<Sn O→Fn'

12	FIL :Sn$^{※1}$ 
	(Filter)
		When Fn= 0, 0→NA', Rn'
		When Fn= 1, Rn+(NA−Rn) ／Sn→NA', Rn'
			28	PWM : Sn$^{※4}$ 
				(Pulse width modulator)
					When NA≦ 0, 0→Rn', 0→Fn'
					When NA> 0, Rn+ 1→Rn' and
					if Rn' <Sn/2·NA/1000, 1→Fn',
					if Rn' ≧Sn/2·NA/1000, 0→Fn,
					if Rn' ≧Sn, 0→Rn

13	INT : Sn$^{※2}$  (Integrator)	When Fn= 0, 0→NA', Rn' When Fn= 1,Rn+NA/Sn→NA',Rn'	29	WP : xi 	BA→xi
14	DIF : Sn$^{※3}$  (Differentiator)	When Fn= 0, 0→NA', Rn' When Fn= 1, (NA−Rn)Sn→NA, NA→Rn'	30	WN : xi 	$\overline{BA}$→xi
15	NOP	No operation	31	EXP : n	EXPANDER

Notice :
- NA, BA : Contents of NA, BA before operation
- NA', BA' : Contents of NA, BA after operation
- Xn : Nn, Sn or Mn can be used, and only one address can be specified.
- Xi : Nn, Sn or Mn can be used, and multiple addresses can be specified.
- Rn, Mn, Sn, Rn, Fn : Only a specified address can be used, and only one address can be specified.
- xn : B, F or H address can be used, and only one address can be specified.
- xi : B, F or H address can be used, and multiple address can be specified.

※1 Time constant/sampling period
※2 Integral time/sampling period
※3 Differential time/sampling period
※4 Pulse period/sampling period



Operation word and address word are discriminated by the code of bits 1,2

Fig. 4 Format of macro-instructions

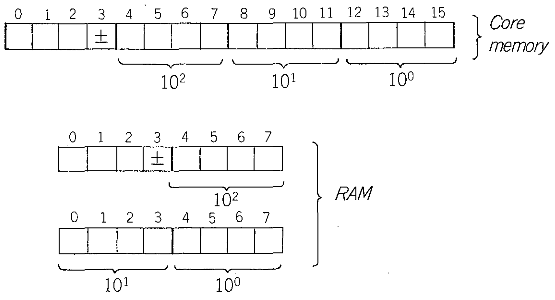


Fig. 5 Format of numerical data

(1) Specific data can be automatically transmitted by specifing the start of transmission with the program.

(2) The transmission mode is asynchronous and a transmission speed matched to the opposite apparatus can be obtained.

(3) Since parity and repetitive transmission checks are performed, the reliability of the transmission data is high.

VI. PROGRAMMING SYSTEM FOR THE ARITHMETIC MODULE

Macro-instructions based on block diagram expression are available for the application program.

The internal operations to execute the macro-instructions are all performed by the control program described by 8080 instructions in the processing unit of the arithmetic module in *Fig. 4*. Therefore, the actual application program can be described by macro-instructions and stored in the program memory.

The airthmetic module operates in accordance with the program described by the macro-instructions.

A macro-instruction is composed of 8 bits and classified as operation words and address words as shown in the format of *Fig. 5*.

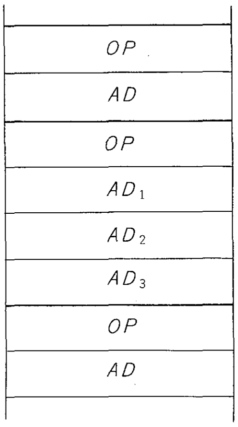
Both numerical and on/off data are processed by the arithmetic module.

The numerical data format is shown in *Fig. 6*. The addresses of the data are discriminated in accordance with the application shown in *Fig. 7*.

Classification	*Numerical data*		*ON/OFF data*	
	SC Core memory	*RAM*	*SC Core memory*	*RAM*
1 2 0 0	*Nn*		*Bn*	
1 2 0 1	*Sn*	*Rn*	*Fn*	
1 2 1 0		*Mn*		*Hn*

○ n=0~31 *specified with address word*
○ *Numerical accumulator (NA)=Mo*
○ *Binary accumulator (BA)=Ho*

Fig. 6 Data symbols and designation



OP: *Operation word*
AD : *Address word*

Fig. 7 Construction of program

Nn and Sn are the external data addresses located on the data area of the core memory. Nn is mainly used for the external input/output.

Sn is used for setting input. Rn and Mn are on the data memory (RAM) of the arithmetic module. Rn is mainly used as the auxiliary resistor of Sn, and Mn is used as the internal data memory. Bn and Fn are the external flags located on the data area of the core memory. Bn is used as the input/output signal for the arithmetic module.

Fn is mainly used as the auxiliary flag of Sn.

The flag Hn is in the data memory of the airthmetic module and is used as the internal flag for logic operation. In addition the numerical accumulator (NA) is located at Mo (n=0, on Mn) and the bit accumulator (BA) is located at Ho (n=0, on Hn).

The macro-instruction table is given in *Table 3*.

One macro-instruction consists of one operation word and one or more address words. Application programs are described with macro-instructions. The processing unit reads the macro-instruction, decodes the operation word and manipulates the data or flag specified by the address words. In the case of an external address (Nn, Sn, Bn or Fn), the core memory is operated through the interface. In the case of an internal address (Mn or Hn), the data memory (RAM) in the module is operated directly.

The application program is started by an interval timer. The interval is in n multiples of 50 ms and n can be specified from 0 to 255.

Consequently, the program start interval can be ar-

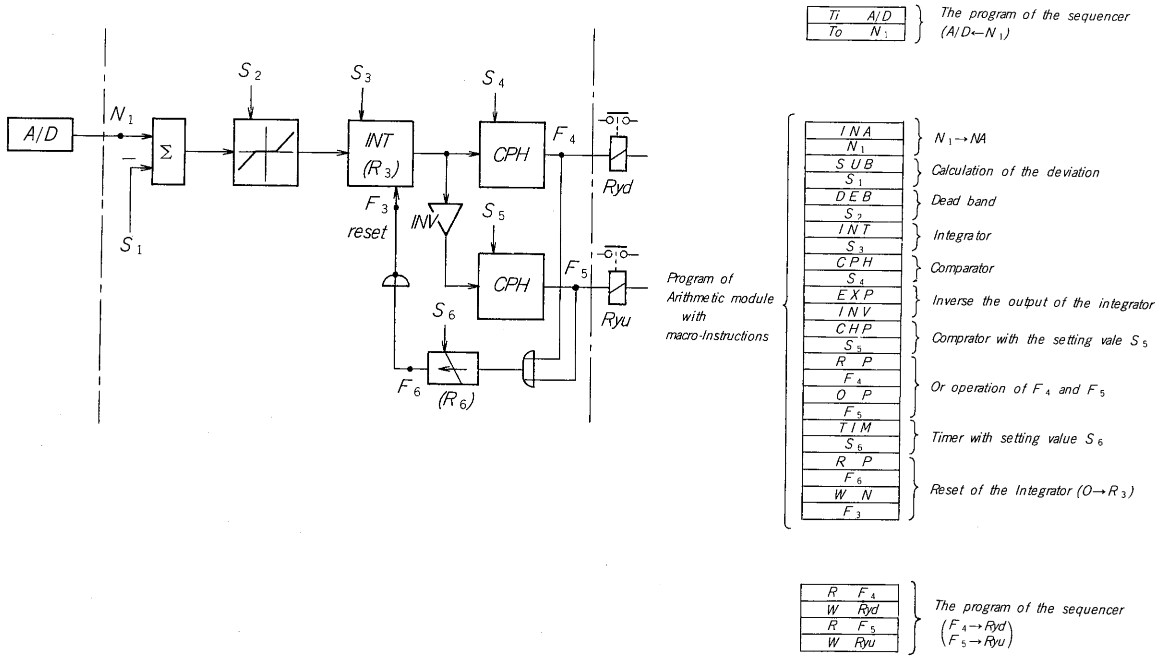

A/D
N_1
S_1
S_2
S_3
S_4
S_5
S_6
Σ
INT
(R_3)
F_3
reset
INV
CPH
CPH
F_4
F_5
F_6
(R_6)
Ryd
Ryu
Ti A/D
To N_1
The program of the sequencer ($A/D \leftarrow N_1$)
Program of Arithmetic module with macro-Instructions
INA
N_1
$N_1 \rightarrow NA$
SUB
S_1
Calculation of the deviation
DEB
S_2
Dead band
INT
S_3
Integrator
CPH
S_4
Comparator
EXP
INV
Inverse the output of the integrator
CHP
S_5
Comprator with the setting vale S_5
R P
F_4
O P
F_5
Or operation of F_4 and F_5
TIM
S_6
Timer with setting value S_6
R P
F_6
W N
F_3
Reset of the Integrator ($O \rightarrow R_3$)
R F_4
W Ryd
R F_5
W Ryu
The program of the sequencer ($F_4 \rightarrow Ryd$, $F_5 \rightarrow Ryu$)

Fig. 8 Example of program with macro-instructions

bitrarily set over the 50~12,750 ms (12.75 sec.) range by 50 ms. The program may also be started by the sequencer program.

Five application programs can be registered in the program memory of the arithmetic module.

Each program can be individually started at the desired intervals.

Therefore, the programs can be divided into high speed control programs and low speed programs. In addition, the location of the flag and data address area for each program can be individually specified.

The above time intervals and address areas are defined by the control table in the program memory.

The arithmetic module uses the data on the core memory of the sequencer and manipulates them in accordance with the application program of the module. The data transfer between the core memory (data area) and the I/O modules (including A/D & D/A module) is performed by means of the sequencer program. Therefore, the sequencer incorporating the arithmetic module also acts as the I/O interface of the module.

An example of an application program is shown in *Fig. 8.*

VII. CONCLUSION

The basic composition of the sequencer in simple and functions can be added to meet requirements. This is intended to achieve a functional expansion system which will result in a control system with high cost performance.

It is hoped that this article will serve as a reference in the rational design of control systems including this Fuji sequencer.