DESCRIPTION

System Design Support Equipment

Field of the Invention

This invention relates to a system design support equipment for supporting a software making (programming) or a design necessary for a system development.

Background Art

An operational processor used for controlling systems such as plants consists of a plurality of elements (hereinafter, refers as "functional block") that realize fundamental functions such as addition and multiplication or any combination thereof (referred as "complex function"). These elements contain not only programs to realize a required function but also a circuit that is a hardware. These elements can be understood as "components" or "modules". Particularly in a field of software technology these components are called as "object components" because they belong to a general idea of "object". Combining several object components makes an object component group that has a specific function (referred as "complex object component"). The complex object component can be expressed as complex functional block that is in higher layer relatively to a plurality of functional blocks in lower layer. Also, in some cases, combining complex object components can make a higher layer complex function object.

There have been proposed the following design support software for creating a complex object component by combining the object components mentioned above.

(1) "LabVIEW" of U.S. company "National Instruments"

This is a programming support tool in which various object components (so-called "Virtual Instrument", hereinafter, expressed as "VI") are prepared as algorithm such as numerical operation, and a certain application function can be built with optional VI group and linkages of the VI's. According to the system so-called linking diagram editor, grouping suitable parts expressed with linkages of a plurality of blocks can make a complex object component (so-called "sub VI") including them. The complex object component can be used in the same application and the other applications by duplicating

the complex object component. In this case, each place where the complex object component is used refers to the object component group included in lower layer than the original complex object component (the object component group is commonly used by all places for use).

When the complex object component (sub VI) is used in several situations, information is commonly used in both of the original (source) and user sides. There is no problem if the information of object component group in lower layer may be completely same in the original and the user sides, but such situation is rare. In many cases, for utilizing the complex object component, it is required to change attribute information of the components in lower layer, or to replace the object components in lower layer with the other kind of object components. In such cases, it is necessary to copy whole contents of the object component group in lower layer to prepare the complex object component separately and utilizing it with changing the required portion. In other words, it is necessary to prepare a large number of the complex object components that is changed partly (accordingly, similar as a whole), resulting in reduction in utilization of individual complex object.

(2) " Matlab/Simulink "of U.S. company "Math Works"

This is a similar programming support tool, and enables a complex object component (called as "subsystem") to be produced by almost similar operation. This complex object component can be utilized by being copied in the same application and the other applications. In this case, all the object component groups included in lower layer are duplicated to exist separately in each place for use.

Further, an attribute of a specified component among object component group included in lower layer can be treated as an attribute of complex object component in higher layer. If an attribute value is set in a place where the complex object component is used, then it is set as an attribute value of the corresponding component in lower layer.

"Attribute" means an item characterizing appearance, condition and the other properties of an object. In case of the functional block mentioned above, the "attribute" is an item (for example, various kinds of operation parameters) necessary for executing the function (for example, one of four basic arithmetic operations). Each attribute is

usually followed by additional information to explain the attribute, that is, information such as a default value of the attribute (the value set at first when the functional block is formed) and limitation of the attribute value (the range in which the attribute value can take when changed).

Using the above complex object component (subsystem), it is not necessary to prepare similar complex object component so long as an attribute value is contained in the changeable range, different from the above (1), because the value of specified object component in lower layer can be set from complex object component in higher layer. However, "class" of the specified object component in lower layer cannot be changed. In other words, a certain specified object component in lower layer cannot be changed to an object component having different function. In such case, it is necessary to prepare the other complex object component in which the specified object component has been changed to an object component of the other class.

"Class" is an item indicating a kind or model (type) to distinguish the attribute and function of an object. This corresponds to a class in object-oriented software technology, that is, a template of object which includes variable (attribute value) representing the attribute of an object and function.

When the complex object component is utilized at several points, the object component group in lower layer is entirely copied in every places for use. In other words, a subsystem of the source or origin and a subsystem of each place for use have separate component group, respectively. In this case, if there is an error in information of the object component group in lower layer of the origin and the error is corrected, a similar correction must be made on information of the object component group duplicate in lower layer of the latter subsystem, resulting in extra maintenance works.

Though the duplicate and the origin have the same content, double capacity of memory is required in useless in the computer. Also, when the original and copied information are stored in a memory such as hard disc, the capacity of the hard disc is required in double because these information are stored separately though the content is identical each other. When many copies are required, the extra capacity needed increases in proportion to the number of copies.

Accordingly, an object of this invention is to provide a system design support equipment improved in (re)utilization and workability of maintenance of complex object component when a complex object component having specified function is prepared as a higher layer functional block by combining a plurality of function (containing not only fundamental function but also complex function) block as lower layer elements for system design.

Another object of this invention is to provide a system design support equipment which can reduce memory capacity required in the above mentioned work and capacity of memory device used for storing.

Disclosure of Invention

This invention provides a system design support equipment for producing a complex functional block of higher layer by combining a plurality of functional blocks that realize functions necessary for system design as lower layer elements, which comprises an information providing portion that extracts an attribute from individual functional block of lower layer and shows information to have an operator appoint the attribute to be peculiar attribute in higher layer or to be common information in low and higher layers; an appointment input portion for the operator to appoint the attribute of the individual functional block of lower layer to be the peculiar attribute or to be the common information based on the information shown by the attribute information providing portion; a peculiar information registration processor to register the attribute as peculiar attribute when the operator appointment makes the peculiar attribute; a peculiar information memory to store the peculiar attribute; and a common information memory to store information other than the peculiar attribute as the common information, wherein the peculiar attribute and the common information are made utilizable distinguishably in the complex functional block of higher layer.

According to this invention, when a functional block of higher layer having a similar attribute in the same structure as combination of the functional blocks of lower layer is prepared, it is possible to produce only minimum substance needed and to set the attribute of the functional block of lower layer to be established individually to a desired

attribute, correspondingly to each similar functional block of higher layer, without increasing combination structure itself of the functional blocks of lower layer more than that required as substance of computer program (substance portion). Therefore, all information of lower layer other than the information established as the peculiar attribute may be common in a plurality of functional blocks of higher layer, resulting in reduction of the capacity of memory necessary for additional new preparation of similar functional blocks.

Also, regarding maintenance of the attributes of functional blocks of lower layer that are common in a plurality of similar functional blocks of higher layer, the maintenance work is completed by modification in the combination of the lower layer functional blocks, resulting in improvement in efficiency of software maintenance work.

An embodiment of this invention is constituted so that the information providing portion extracts a class for distinguishing function of the individual functional blocks of lower layer, and shows information to have an operator appoint the class to be a peculiar attribute in higher layer or to be a common information in low and higher layers. Based on the information shown by the information providing portion, the operator appoints the class of the individual functional blocks of lower layer to be peculiar attribute or to be common information, and when the appointment indicates the peculiar attribute, the peculiar information registration processor registers the class as peculiar attribute and the common information memory stores the information other than the peculiar attribute, and thus the information other than registered as the peculiar attribute can be commonly used in the complex functional block of higher layer.

According to the embodiment mentioned above, on preparation of a higher layer complex functional block having similar attribute and component in the same structure of combination of lower layer functional blocks, without increasing combination structure itself of functional blocks of lower layer as substance of computer program more than required, only minimum necessity of substance may be created and besides, an attribute of functional block of lower layer to be established individually to each similar functional block of higher layer can be established as desired attribute. As a result, as an attribute of higher layer functional block, it is possible not only to change an attribute

value of a certain specified functional block in lower layer to a peculiar value, but also to change a class of the specified lower layer functional block, so that a higher layer functional block having wider utility can be provided..

Also, all information of attribute and class of lower layer other than the information established as peculiar attribute can be common in a plurality of functional blocks of higher layer, resulting in reduction of capacity of memory necessary for additional new preparation of similar functional blocks.

Further, regarding maintenance of attribute and class of lower layer functional blocks that are common in a plurality of similar functional blocks of higher layer, maintenance work is completed by modification in combination of lower layer functional blocks, resulting in improvement in efficiency of software maintenance work.

According to a specified embodiment, a lower layer functional block includes a complex functional block constituted by combining the other functional blocks. Also, a lower layer functional block may include a fundamental functional block realizing minimum unit of fundamental function.

According to another specified embodiment, a class of lower layer functional block includes groups of class corresponding to various kinds of arithmetic functions in response to input. In this case, arithmetic parameter that is an attribute of a certain specified functional block among the lower layer functional blocks can become an attribute of higher layer complex functional block. Also, a higher layer complex functional block may include an attribute for changing appointment of a class of a certain specified functional block in lower layer to a class corresponding to the other arithmetic.

According to another specified embodiment, a class of lower layer functional block includes a functional class that outputs a constant. In this case, a value of constant that is an attribute of the functional block of lower layer can be included as one of the attribute of higher layer complex functional block.

According to another specified embodiment, a class of lower layer functional block includes a class that defines a relation between a plurality of inputs and a plurality of outputs. The relation defining class includes parallel connection class in which two inputs are output as they are, respectively and cross connection class in which two inputs

are output with being replaced each other. A higher layer complex functional block can have an attribute to appoint with changing a lower layer functional block of parallel connection class to a functional block of cross connection class, or to appoint with changing a lower layer functional block of cross connection class to a functional block of parallel connection class.

As mentioned above, according to this invention, "attribute" or "class" of lower layer functional block is registered as peculiar information in higher layer or common information in higher layer and lower layer, so that the peculiar information and the common information can be utilized distinguishably each other in higher layer functional block. That is, higher layer functional block registers only peculiar information and, as to common information if necessary, higher layer refers to (common) information stored in lower layer. Consequently, it is possible to realize an improvement in utility of complex functional object component, an improvement in workability of maintenance such as modification of complex object component and reflection to place for use thereof, reduction in capacity of memory required and reduction in capacity required to a secondary memory device such as hard disc or the like necessary for permanent store of information.

Brief Description of Drawings

Fig. 1 is a block diagram showing constitution of one embodiment of this invention;

Fig. 2 shows appearance of a personal computer set used as system design support equipment of this invention;

Fig. 3 shows examples of fundamental functional block used in this invention;

Fig. 4 shows an example of complex functional block and lower layer components;

Fig. 5 shows a complex functional block to change class of lower layer functional blocks, and lower layer components included therein;

Fig. 6 shows a complex functional block in which two inputs and two outputs are connected in "PARALLEL", respectively, and lower layer components included

therein;

Fig. 7 shows a complex functional block in which two inputs and two outputs are connected in "CROSS", respectively, and lower layer components included therein;

Fig. 8 shows a complex functional block in which functional block of Fig. 5 is added by functional block of Fig. 6, and lower layer components included therein;

Fig. 9 shows an example of screen display of system design of this invention;

Fig. 10 shows an example of "functional sheet" screen display for defining complex functional block by means of a plurality of functional blocks;

Fig. 11 shows an example of "input and output" definition screen display for defining input and output of complex functional block;

Fig. 12 shows an example of "icon" definition screen display for defining icon that shows complex functional block;

Fig. 13 shows an example of peculiar attribute registration screen display for registering attribute of functional block that is component of complex functional block as "peculiar attribute";

Fig. 14 shows an example of peculiar attribute registration confirmation screen display for confirming that one registered as peculiar attribute has been registered as attribute of complex functional block;

Fig. 15 shows an example of complex functional block utilization screen display for utilizing complex functional block for system under design by disposing them on the screen; and

Fig. 16 shows an example of peculiar parameter preset screen display for presetting parameter of peculiar attribute in each place where the complex functional block is used.

Best Mode For Carrying Out The Invention

Fig. 1 is a block diagram showing the constitution of one embodiment of the system design support equipment according to this invention. This equipment has a plurality of functional blocks realizing functions required for system design as lower layer components, and creates higher layer functional blocks by combining them. As

means to realize the operation, for example, a personal computer system 1 shown in Fig. 2 is used. The personal computer system 1 comprises a main body 2 accommodating memory such as CPU and hard disc, as generally known as hardware, CRT (or liquid crystal) display 5 and printer 6 as output means, keyboard 3 and mouse 4 as input means, and FDD (floppy disc driver) 7 and CDD (CD-ROM driver) 8 as input and output means. Constitution as the system design support equipment according to this invention is divided into respective sections having following functions:

① An appointment input portion 11 for an operator to appoint "attribute" of the individual functional block of lower layer and "class" for distinguishing its function to be the peculiar attribute or to be the common information. This includes keyboard 3 and mouse 4 in the computer system of Fig. 2.

② An information providing portion 12 that extracts an attribute from individual functional block of lower layer and shows information to have an operator appoint the attribute to be "peculiar" attribute in higher layer or to be "common" information in lower and higher layers. This includes CPU in the main body 2 and display 5 shown in Fig. 2.

③ A peculiar information registration processor 13 to register the attribute as peculiar attribute, when the appointment by the operator is made to "peculiar" attribute. This consists of CPU in the main body 2.

④ A peculiar information memory 14 to store peculiar information (such as kind of attribute, default value of each attribute or the like). This consists of memory in the main body 2.

⑤ A common information memory 15 to store information other than peculiar attribute, as common information. This consists of memory in the main body 2.

A usable higher layer complex functional block 16 is constituted by combining information stored in ④ and ⑤ mentioned above.

On actuation of this equipment, the operator can operate the appointment input portion 11 to appoint the operator to appoint attributes of the individual functional blocks of lower layer and class to be the peculiar attribute or to be the common information. Responding to the appointment operation, this equipment distinguishes between peculiar

information and common information in higher layer complex functional block 16 so that the information can be used again for preparing new complex functional blocks 21 and 22. That is, the newly prepared complex functional blocks 21 and 22 correspond to objects created from the class.

As shown in Fig. 1, the higher layer complex functional blocks 21 and 22 contain peculiar information memory 31 and common information viewer 32, respectively. The peculiar information memory 31 stores a copy of content of the peculiar information memory 14 of utilizable complex functional block 16. On the other side, the common information viewer 32 is a portion (pointer preset in memory) for referring to content of the common information memory 15, if necessary.

Examples are explained below.

Fig. 3 shows examples of "fundamental functional block" (functional block of minimum unit) such as the four basic operations of arithmetic. In the figure, (A) is "INPUT" functional block IN showing input (or input terminal), (B) is "OUTPUT" functional block OUT showing output (or output terminal), (C) is "CONSTANT" functional block CONST outputting constant, (D) is "ADDITION" functional block ADD to carry out addition of two inputs and output the result, (E) is "SUBTRACTION" functional block SUB to carry out subtraction of two inputs and output the result, (F) is "MULTIPLICATION" functional block MUL to carry out multiplication of two inputs and output the result, and (G) is "DIVISION" functional block DIV to carry out division of two inputs and output the result. The classes of these "fundamental functional blocks" are input, output, constant output, addition, subtraction, multiplication and division, respectively.

Fig. 4 shows an example in which "complex functional block" is constituted by combining the "fundamental functional blocks" as mentioned above. In Fig. 4, (A) shows 2-inputs & 1-output typed arithmetic unit to carry out multiplication of two inputs x, y by coefficients A1, A2, respectively, and to output sum of the obtained amounts, as a higher layer complex functional block (complex object component). (B) shows input functional blocks IN1, IN2 representing two inputs x, y; two constant functional blocks CONST (constant values A1, A2); two multiplication functional blocks MUL to obtain

products of input x, y by constant values A1, A2, respectively; an addition functional block ADD to obtain sum of the products; and an output functional block OUT to output the sum, as lower layer functional blocks (lower layer object components) for creating the complex functional block of (A).

In the above example, constant values A1, A2 (each default value equals 1) of constant functional block CONST of (B) can be "attribute" of higher layer functional block of (A), and the function is to output a result of addition of a product of input x by attribute A1 and a product of input y by attribute A2.

In constitution example of Fig. 4, complex functional block of "A1x +A2y" is defined as above. Also, when the functional components performing other operations, that is, A1x−A2y, A1x×A2y, A1x÷A2y are required, complex functional block for each arithmetic can be defined in the same way. However, it is not effective each time to define complex functional block.

Therefore, complex functional block of "A1x (operation) A2y" is defined based on the complex functional block of "A1x +A2y" as shown in Fig. 5. Here, it is possible to appoint the change from an adder block ADD of lower layer to any functional block of other classes of subtraction, multiplication, division in an attribute "operation" of complex functional block of higher layer. In this case, for example, the attribute value may be information to show any of classes of the addition (ADD), subtraction (SUB), multiplication (MUL), and division (DIV), and also the attribute value may be, for example, rows of letter (ADD, SUB, MUL, DIV, etc.) or numerical values. As the result, there can be obtained such effects that the other complex functional blocks that carry out operation can be created with high efficiency, and that utility range of complex functional block spread. Also, in the place where the complex functional block of "A1x +A2y" has already been used, there is no necessity to provide any special change because a default value of attribute of this "operation" is determined as ADD.

Figs. 6 and 7 show examples of the other complex functional blocks, respectively. Fig. 6(A) shows a parallel connector (PARALLEL) as higher layer functional block that has two inputs IN1, IN2 made to outputs OUT1, OUT2 as they are, and in lower layer functional block, as shown in Fig. 6(B), two inputs IN1, IN2 are

connected to two outputs OUT1, OUT2 in parallel. Fig. 7(A) shows a cross connector (CROSS) as complex functional block of higher layer that has two inputs IN1, IN2 replaced mutually, and made to outputs OUT1, OUT2. As shown in Fig. 7(B), two inputs IN1, IN2 are intersected, and internally connected to two outputs OUT1, OUT2.

By using complex functional blocks of Figs. 6(A) and 7(A), utility range of complex functional block of "A1x (operation) A2y" can be widened more. In other words, as shown in Fig. 8(A), in lower layer (B) of complex functional block (A) of "(input switching) A1x (operation) A2y", the "PARALLEL" functional block is disposed just after two inputs IN1, IN2, and can be replaced by the "CROSS" functional block of another class in "input switching" attribute (attribute value = PARALLEL or CROSS) of higher layer functional block. As the result, complex functional block to carry out operation as mentioned above can be created with high efficiency, and besides, utility range of complex functional block of "A1x (operation) A2y" can be widened.

In fact, when the "input switching" functional block is "PARALLEL" in Fig. 8(B), OUT = A1x +A2y if operation = ADD, OUT = A1x - A2y if operation = SUB, OUT = A1x×A2y if operation = MUL, and OUT = A1x÷A2y if operation = DIV. When the "input switching" functional block is "CROSS", OUT = A1y+A2x if operation = ADD, OUT = A1y - A2x if operation = SUB, OUT = A1y×A2x if operation = MUL, and OUT = A1y÷A2x if operation = DIV.

Accordingly, so long as inputs x, y are expressed by usual scalar, even if the "input switching" functional block is any one of "PARALLEL" and "CROSS", results of addition (ADD) and multiplication (MUL) are same. Consequently, "input switching" in complex functional block of Fig. 8(A) is effective when operation is subtraction (SUB) or division (DIV). However, when two inputs x, y are things that vary result of addition or multiplication by their order (whether first term or second term), for example, vector, matrix or quantity expressed by irreversible data, results of addition (ADD) and multiplication (MUL) vary whether the "input switching" functional block is "PARALLEL" or "CROSS". Therefore, even if any of four kind operations is carried out, input switching in complex functional block of Fig. 8(A) is meaningful.

Figs. 9 to 16 show examples of screen display represented on display 5 when

system design is carried out by using the personal computer 1 of Fig. 2 as an apparatus realizing this invention.

First, a system under design is indicated on "system design display" screen as shown in Fig. 9. The system as shown includes a fluid circuit consisting of fluid elements such as control valves and supply lines such as ducts, and a controller that produces signals (two outputs) for controlling action of the fluid circuit in response to two input signals from the fluid circuit. The fluid circuit produces a plurality of outputs (out1, out2, ...) for one fluid input (in) under control by signal from the controller.

In design work of such a system, functional blocks are utilized in the following procedures by the equipment of this invention.

As mentioned above, for creating a complex functional block having function that adds A1 times the input IN1 and A2 times the input IN2 among two inputs and outputs at OUT (the case of Fig. 4 (A)), the complex functional block (A1x+A2y) is defined using basic functional blocks (A) to (D) shown in Fig. 3 on the "function seat" screen display as shown in Fig. 10. As shown in Fig. 9, the basic functional blocks of (A) to (G) of Fig. 3 can be indicated on the display screen together with "system design screen display" or the like as "palette" P storing usable functional blocks. The functional blocks in the palette P correspond to a group of classes of the functional block utilizable at the time. In other words, not only the basic functional blocks (classes) but also complex functional block (class) defined later can be represented in the palette.

Then, in "input and output" definition screen display shown in Fig. 11, input and output of complex functional block are defined. However, in this example, existence of 2-input & 1-output may only be confirmed.

In "icon" definition screen display shown in Fig. 12, an icon symbolizing complex functional block is defined. In this example, since an icon of "rectangle with shadow" is formed automatically as the functional block having 2-input and 1-output, an operator may confirm the icon.

By definitions of complex functional block up to now, the contents are stored entirely in the common information memory 15 of Fig. 1. The "rectangle with shadow" icon representing the complex functional block of this example is indicated by the

"palette" (P of Fig. 9). The indication contents (icon as common information) are referred, each time required, by the common information viewer 32 of the created complex functional block 21 or 22.

Then, in "peculiar attribute registration" display shown in Fig. 13, regarding each of constant functional blocks CONST which are constitution elements of complex functional block (A1x+A2y), the parameter (value of constant) that is an attribute is registered as the "peculiar attribute" when this complex functional block is used. In this example of display, peculiar attribute is registered by adding check mark to check box of "higher layer peculiar". These peculiar attributes are stored in peculiar information memory 14 of Fig. 1.

Next, in "peculiar attribute registration confirmation" display shown in Fig. 14, it is confirmed that two parameter values registered as peculiar attributes as mentioned above have been registered as attributes of complex functional block (A1x+A2y). On this display, it necessary, it is possible to change parameter name to another suitable one.

Then, as shown in Fig. 15, it is possible to use the complex functional block (A1x+A2y) in a system under design by opening the system indication display under design mentioned above and dragging the complex functional block (A1x+A2y) from "palette" indication position of the left side to a suitable position in the system indication display with mouse 4. In this example, the complex functional block (A1x+A2y) is used at two places as circuit element constituting the inside circuit of the controller. For this purpose, if making click on the position of the controller with mouse 4 in the display of Fig. 15, then a display for preparing inside circuit of the controller appears so that two complex functional blocks can be established at the suitable positions in this display (that is, set to places where they are used).

Last, as shown in Fig.16, a parameter value of "peculiar attribute" is established in each place where the complex functional block is used. This is stored in the peculiar information memory 31 of Fig. 1.

Claims

1. A system design support equipment for creating a complex functional block of higher layer by combining a plurality of functional blocks of lower layer elements that realize functions required for system design, which comprises:

an information providing portion that extracts an attribute from individual functional block of lower layer and shows information to have an operator appoint the attribute to be peculiar attribute in higher layer or to be common information in lower and higher layers;

an appointment input portion for the operator to appoint the attribute of the individual functional block of lower layer to be the peculiar attribute or to be the common information based on the information shown by the information providing portion;

a peculiar information registration processor to register the attribute as peculiar attribute when the appointment by the operator is made to peculiar attribute;

a peculiar information memory to store peculiar attribute; and

a common information memory to store information other than peculiar attribute, as common information, wherein

the peculiar attribute and the common information are usable with distinguishing each other in the complex functional block of higher layer.

2. The system design support equipment according to claim 1, wherein

the information providing portion extracts a class from individual functional block of lower layer and shows information to have an operator appoint the class to be peculiar attribute in higher layer or to be common information in lower and higher layers;

the peculiar information registration processor has the operator appoint, by means of the appointment input portion, the class of the individual functional block of lower layer to be peculiar attribute or to be common information based on the information shown by the information providing portion, the peculiar information registration processor registering the class as the peculiar attribute if the operator appoints the peculiar

* C L M S *

attribute; and

the common information memory stores information other than the peculiar attribute, as common information,

whereby the information other than that registered as the peculiar attribute in the complex functional block of higher layer is to be common.

3. The system design support equipment according to claim 1, wherein the lower layer functional block includes a complex functional block constituted by combining the other functional blocks.

4. The system design support equipment according to claim 1, wherein the lower layer functional block may include a fundamental functional block realizing minimum unit of fundamental function.

5. The system design support equipment according to claim 1, wherein a class of the lower layer functional block includes groups of classes corresponding to various kinds of arithmetic functions in response to input.

6. The system design support equipment according to claim 5, wherein an arithmetic parameter that is an attribute of a certain specified functional block in the lower layer functional blocks is included as one of the attribute of higher layer complex functional block.

7. The system design support equipment according to claim 5, wherein the higher layer complex functional block has an attribute for changing appointment of a class of a certain specified functional block of lower layer to a class corresponding to the other arithmetic.

8. The system design support equipment according to claim 2, wherein the lower layer functional block includes a functional class that outputs a constant.

9. The system design support equipment according to claim 8, wherein a value of constant that is an attribute of the functional block of lower layer is included in attribute of higher layer complex functional block.

10. The system design support equipment according to claim 2, wherein the class of lower layer functional block includes a class that defines a relation between a plurality of inputs and a plurality of outputs.

11. The system design support equipment according to claim 10, wherein the class that defines the relation is any of parallel connection class in which two inputs are outputs as they are and cross connection class in which two inputs are outputs replaced each other, and the higher layer complex functional block has an attribute to appoint changing the lower layer functional block of the parallel connection class to a functional block of the cross connection class, or to appoint changing the lower layer functional block of the cross connection class to a functional block of the parallel connection class.

* C L M S *

Abstract

A system design support equipment produces a complex functional block of higher layer by combining a plurality of functional blocks of lower layer elements that realize functions required for system design. The equipment includes: an information providing portion (12) that extracts an attribute from individual functional block of lower layer and shows information to have an operator appoint the attribute to be peculiar attribute in higher layer or to be common information in lower and higher layers; an appointment input portion (11) for the operator to appoint the attribute of the individual functional block of lower layer to be the peculiar attribute or to be the common information based on the information shown by the information providing portion; a peculiar information registration processor (13) to register the attribute as peculiar attribute when the appointment by the operator is made to peculiar attribute; a peculiar information memory (14) to store peculiar attribute; and a common information memory (15) to store information other than peculiar attribute, as common information, wherein in the complex functional block of higher layer, peculiar attribute and common information are usable with distinguishing each other.

FIG. 1

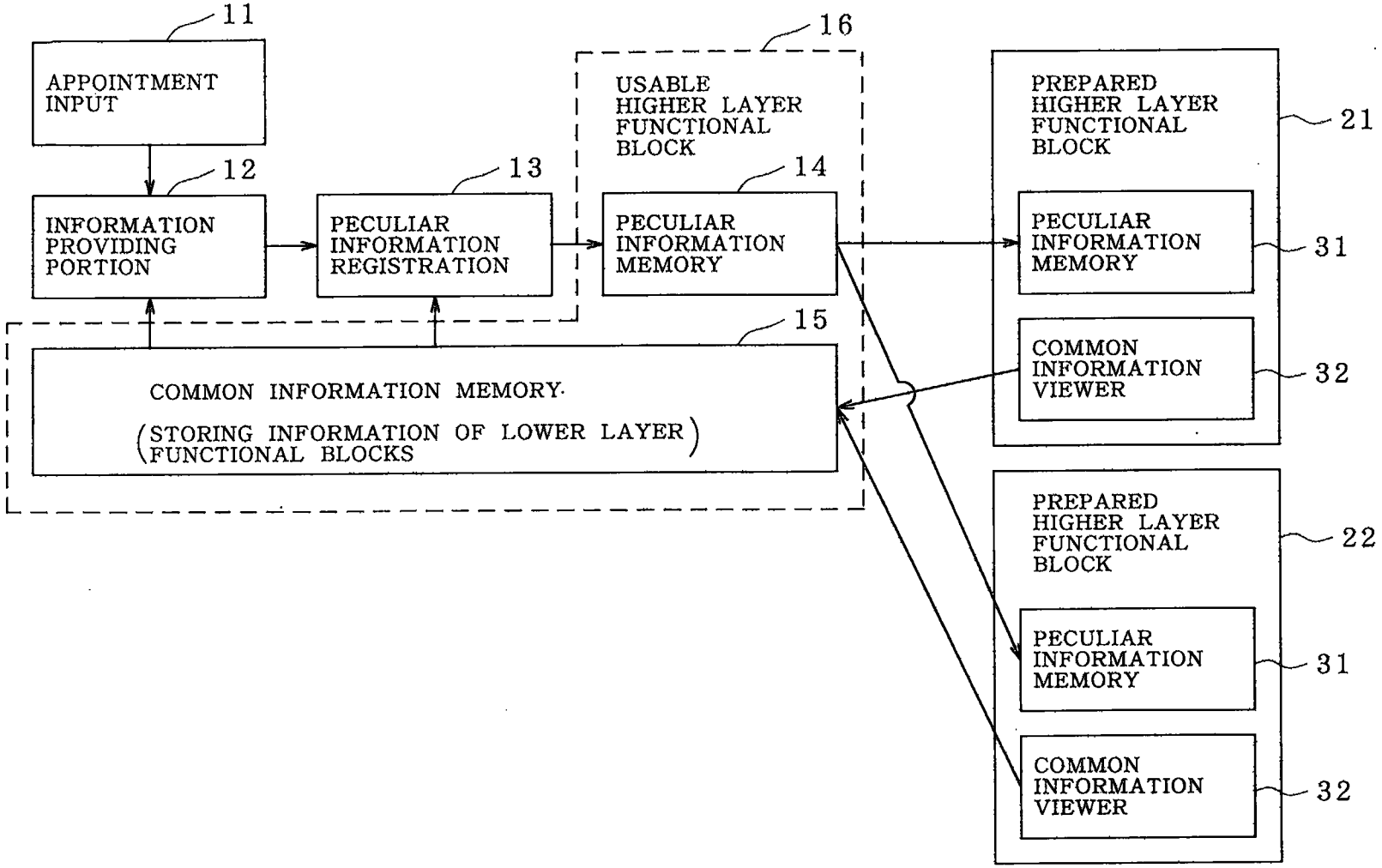

11
APPOINTMENT INPUT
12
INFORMATION PROVIDING PORTION
13
PECULIAR INFORMATION REGISTRATION
16
USABLE HIGHER LAYER FUNCTIONAL BLOCK
14
PECULIAR INFORMATION MEMORY
15
COMMON INFORMATION MEMORY
(STORING INFORMATION OF LOWER LAYER FUNCTIONAL BLOCKS)
PREPARED HIGHER LAYER FUNCTIONAL BLOCK
21
PECULIAR INFORMATION MEMORY
31
COMMON INFORMATION VIEWER
32
PREPARED HIGHER LAYER FUNCTIONAL BLOCK
22
PECULIAR INFORMATION MEMORY
31
COMMON INFORMATION VIEWER
32

FIG. 2

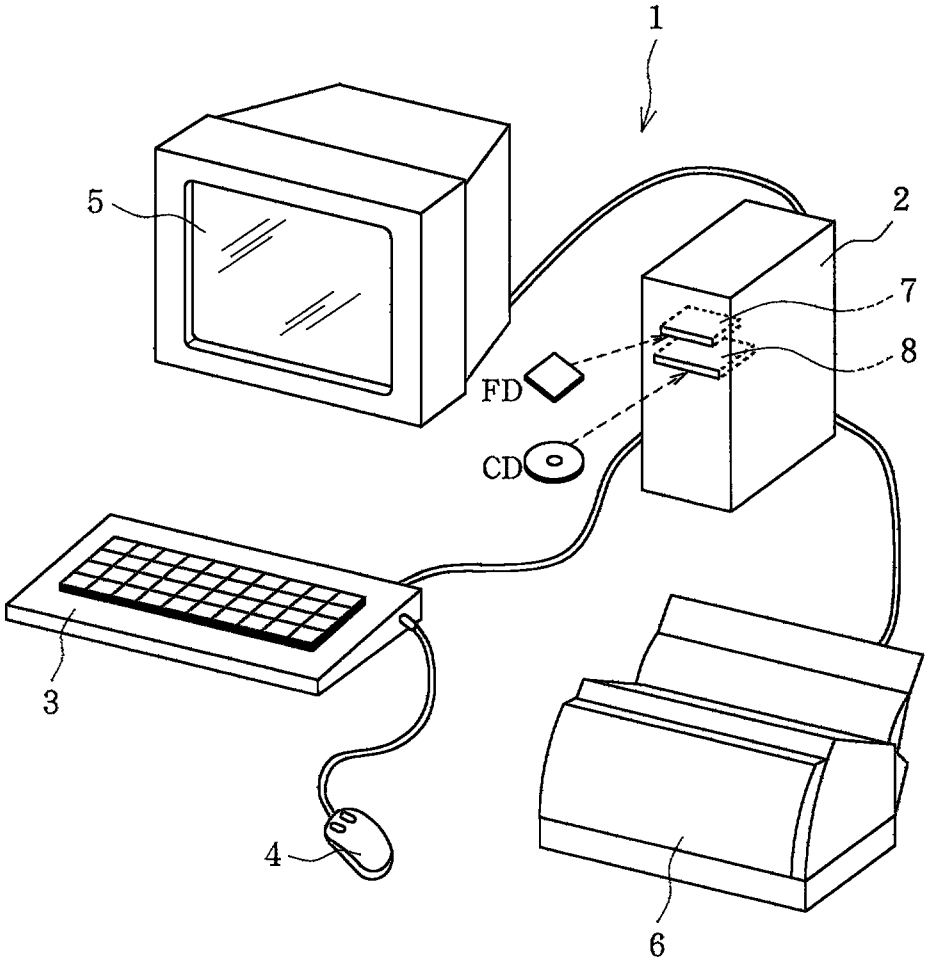


FIG. 3

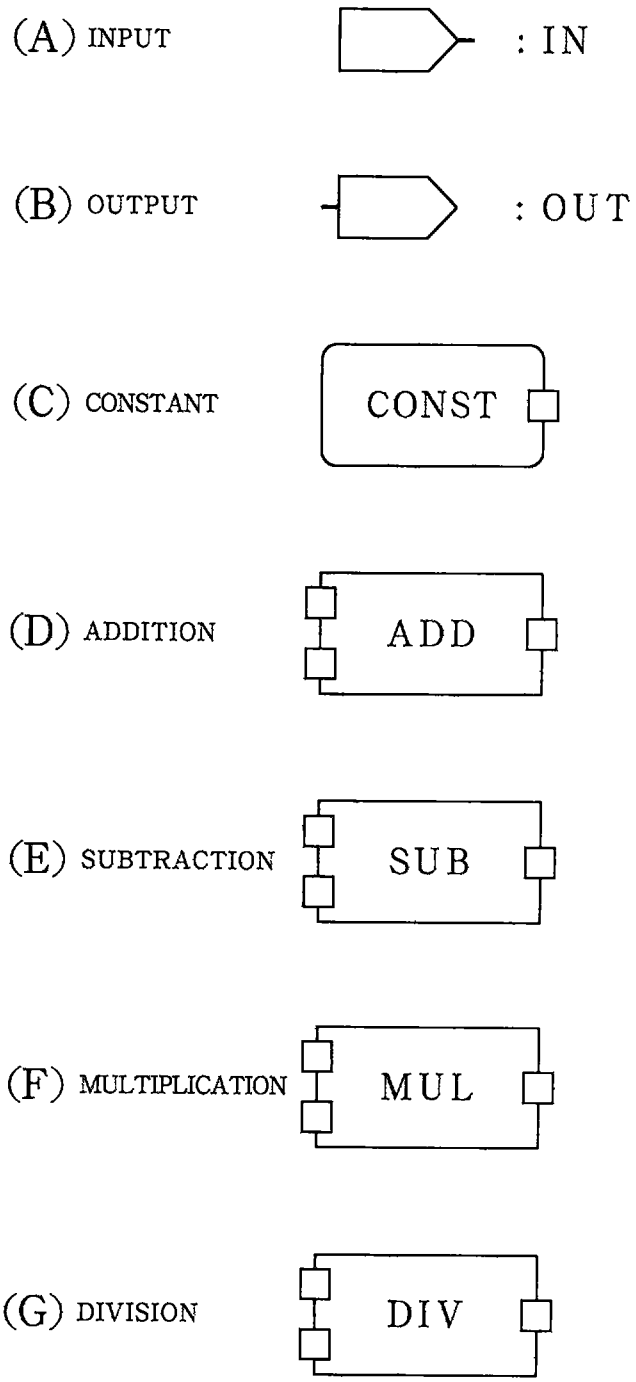

(A) INPUT
: IN
(B) OUTPUT
: OUT
(C) CONSTANT
CONST
(D) ADDITION
ADD
(E) SUBTRACTION
SUB
(F) MULTIPLICATION
MUL
(G) DIVISION
DIV

FIG. 4

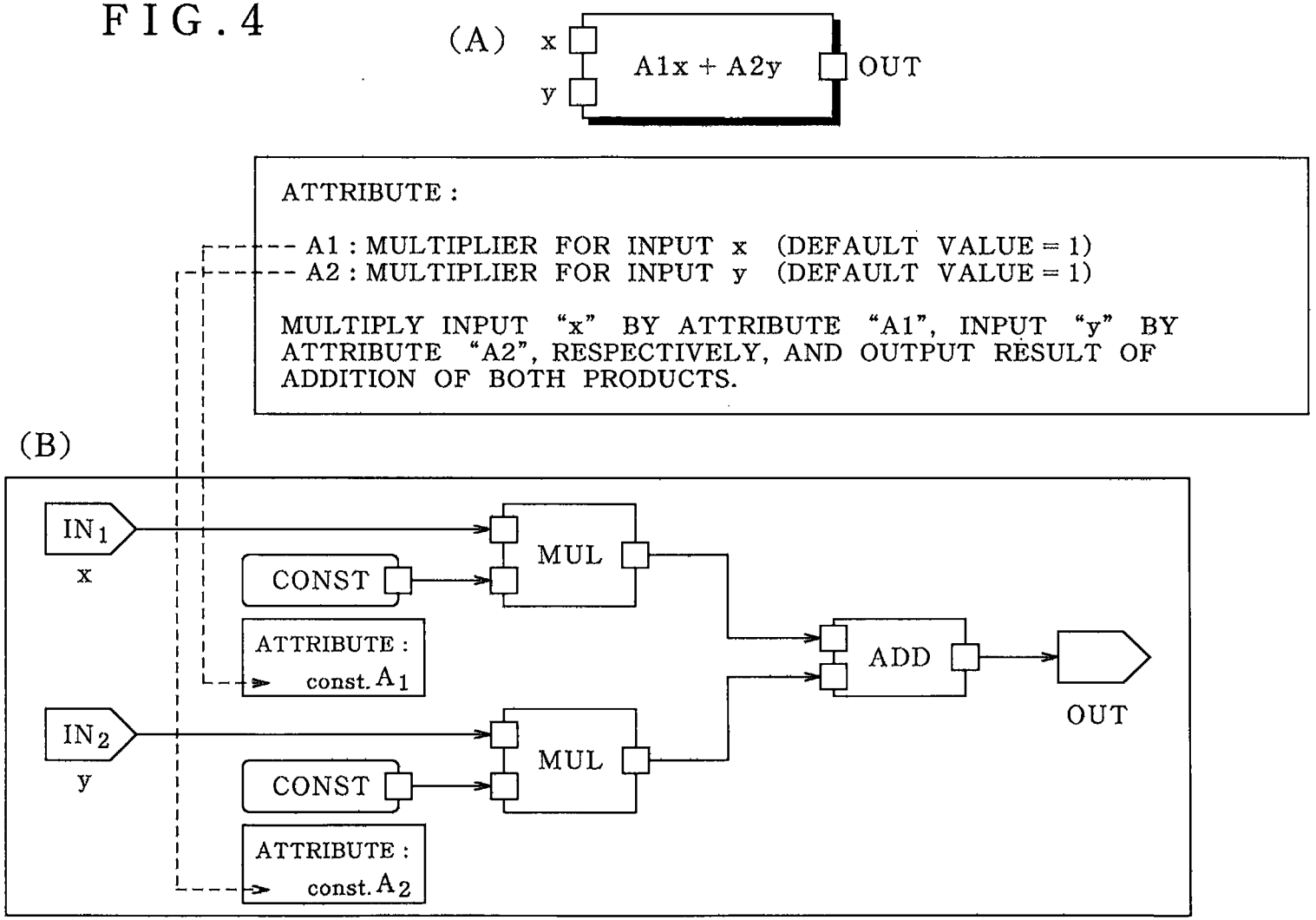

(A)
x
y
A1x + A2y
OUT
ATTRIBUTE :
A1 : MULTIPLIER FOR INPUT x (DEFAULT VALUE = 1)
A2 : MULTIPLIER FOR INPUT y (DEFAULT VALUE = 1)
MULTIPLY INPUT "x" BY ATTRIBUTE "A1", INPUT "y" BY ATTRIBUTE "A2", RESPECTIVELY, AND OUTPUT RESULT OF ADDITION OF BOTH PRODUCTS.
(B)
IN1
x
IN2
y
CONST
CONST
ATTRIBUTE :
const. A1
ATTRIBUTE :
const. A2
MUL
MUL
ADD
OUT

FIG. 5

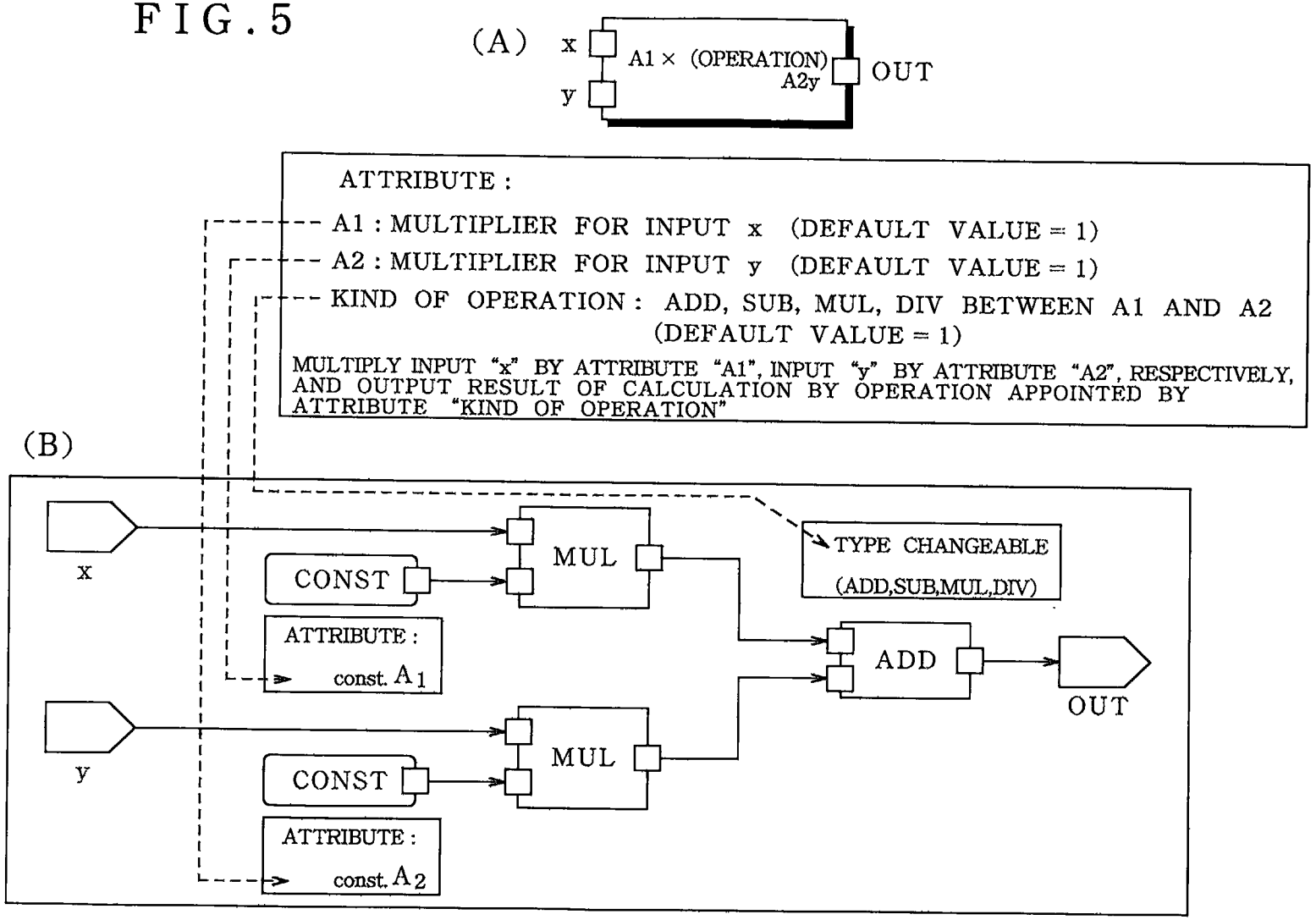

(A)
x
y
A1 × (OPERATION) A2y
OUT
ATTRIBUTE :
A1 : MULTIPLIER FOR INPUT x (DEFAULT VALUE = 1)
A2 : MULTIPLIER FOR INPUT y (DEFAULT VALUE = 1)
KIND OF OPERATION : ADD, SUB, MUL, DIV BETWEEN A1 AND A2 (DEFAULT VALUE = 1)
MULTIPLY INPUT "x" BY ATTRIBUTE "A1", INPUT "y" BY ATTRIBUTE "A2", RESPECTIVELY, AND OUTPUT RESULT OF CALCULATION BY OPERATION APPOINTED BY ATTRIBUTE "KIND OF OPERATION"
(B)
x
y
CONST
CONST
MUL
MUL
ADD
OUT
TYPE CHANGEABLE
(ADD,SUB,MUL,DIV)
ATTRIBUTE :
const. A1
ATTRIBUTE :
const. A2

FIG. 6

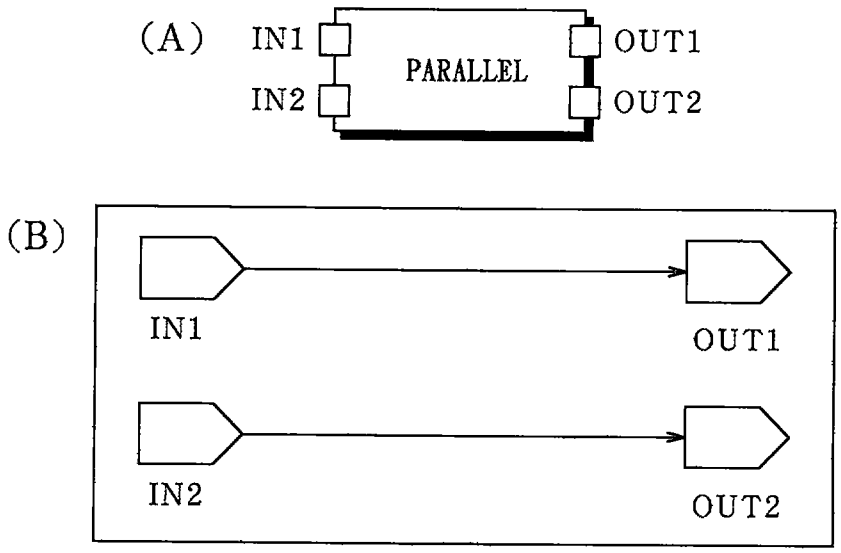

(A)
IN1
IN2
PARALLEL
OUT1
OUT2
(B)
IN1
OUT1
IN2
OUT2

FIG. 7

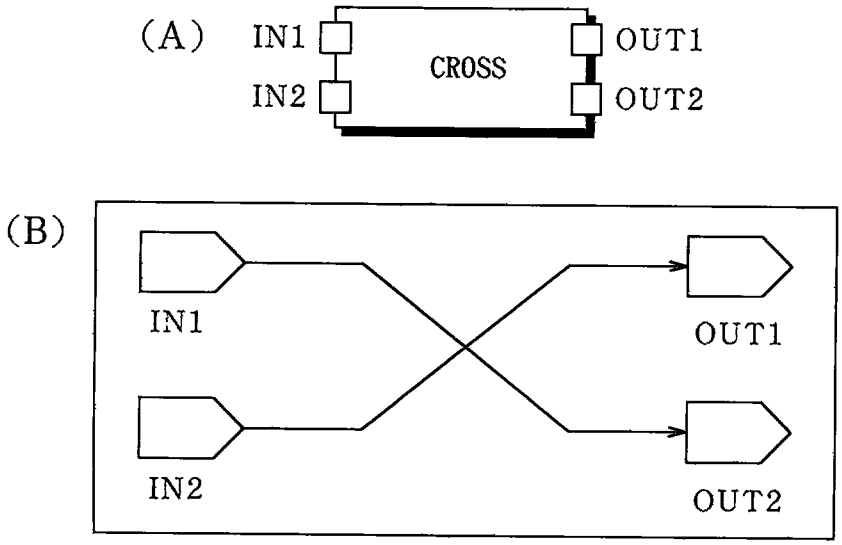

(A)
IN1
IN2
CROSS
OUT1
OUT2
(B)
IN1
OUT1
IN2
OUT2

FIG. 8

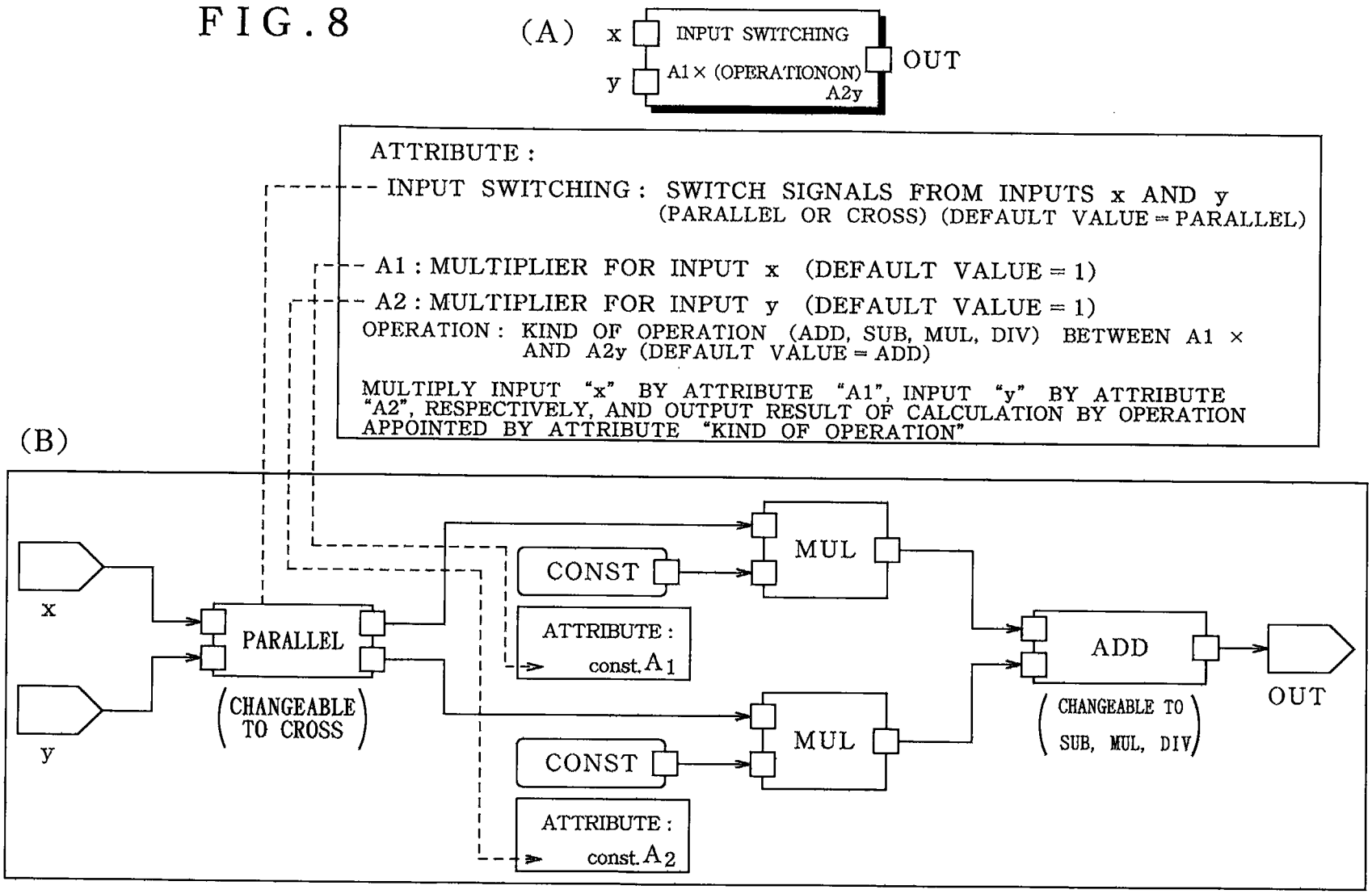

(A)
x
y
INPUT SWITCHING
A1× (OPERATIONON) A2y
OUT
ATTRIBUTE :
INPUT SWITCHING : SWITCH SIGNALS FROM INPUTS x AND y (PARALLEL OR CROSS) (DEFAULT VALUE = PARALLEL)
A1 : MULTIPLIER FOR INPUT x (DEFAULT VALUE = 1)
A2 : MULTIPLIER FOR INPUT y (DEFAULT VALUE = 1)
OPERATION : KIND OF OPERATION (ADD, SUB, MUL, DIV) BETWEEN A1 × AND A2y (DEFAULT VALUE = ADD)
MULTIPLY INPUT "x" BY ATTRIBUTE "A1", INPUT "y" BY ATTRIBUTE "A2", RESPECTIVELY, AND OUTPUT RESULT OF CALCULATION BY OPERATION APPOINTED BY ATTRIBUTE "KIND OF OPERATION"
(B)
x
y
PARALLEL
(CHANGEABLE TO CROSS)
CONST
ATTRIBUTE : const. A1
MUL
CONST
ATTRIBUTE : const. A2
MUL
ADD
(CHANGEABLE TO SUB, MUL, DIV)
OUT

FIG. 9

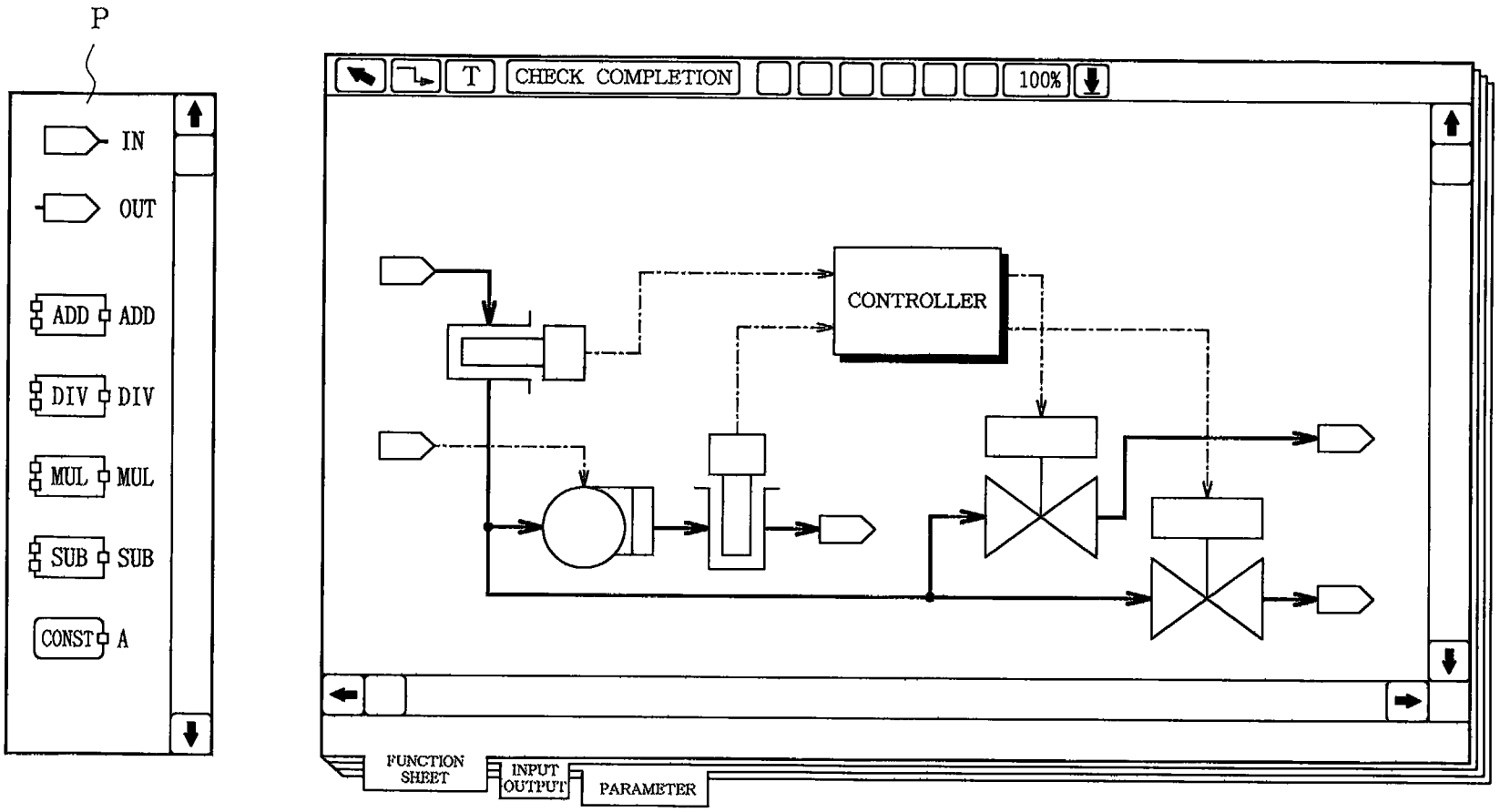

P
IN
OUT
ADD
ADD
DIV
DIV
MUL
MUL
SUB
SUB
CONST
A
T
CHECK COMPLETION
100%
CONTROLLER
FUNCTION SHEET
INPUT OUTPUT
PARAMETER

FIG. 10

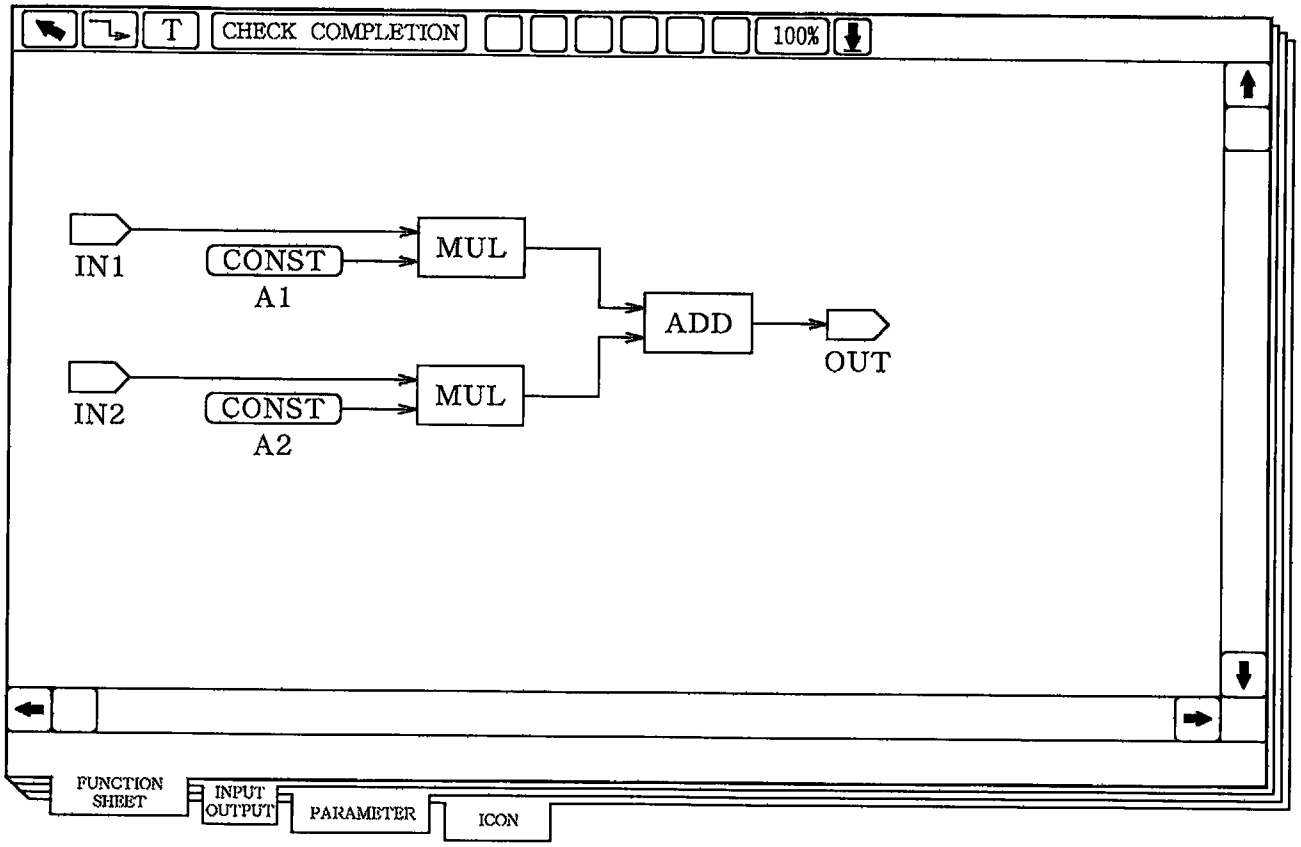

T
CHECK COMPLETION
100%
IN1
CONST
A1
MUL
ADD
OUT
IN2
CONST
A2
MUL
FUNCTION SHEET
INPUT OUTPUT
PARAMETER
ICON

FIG.11

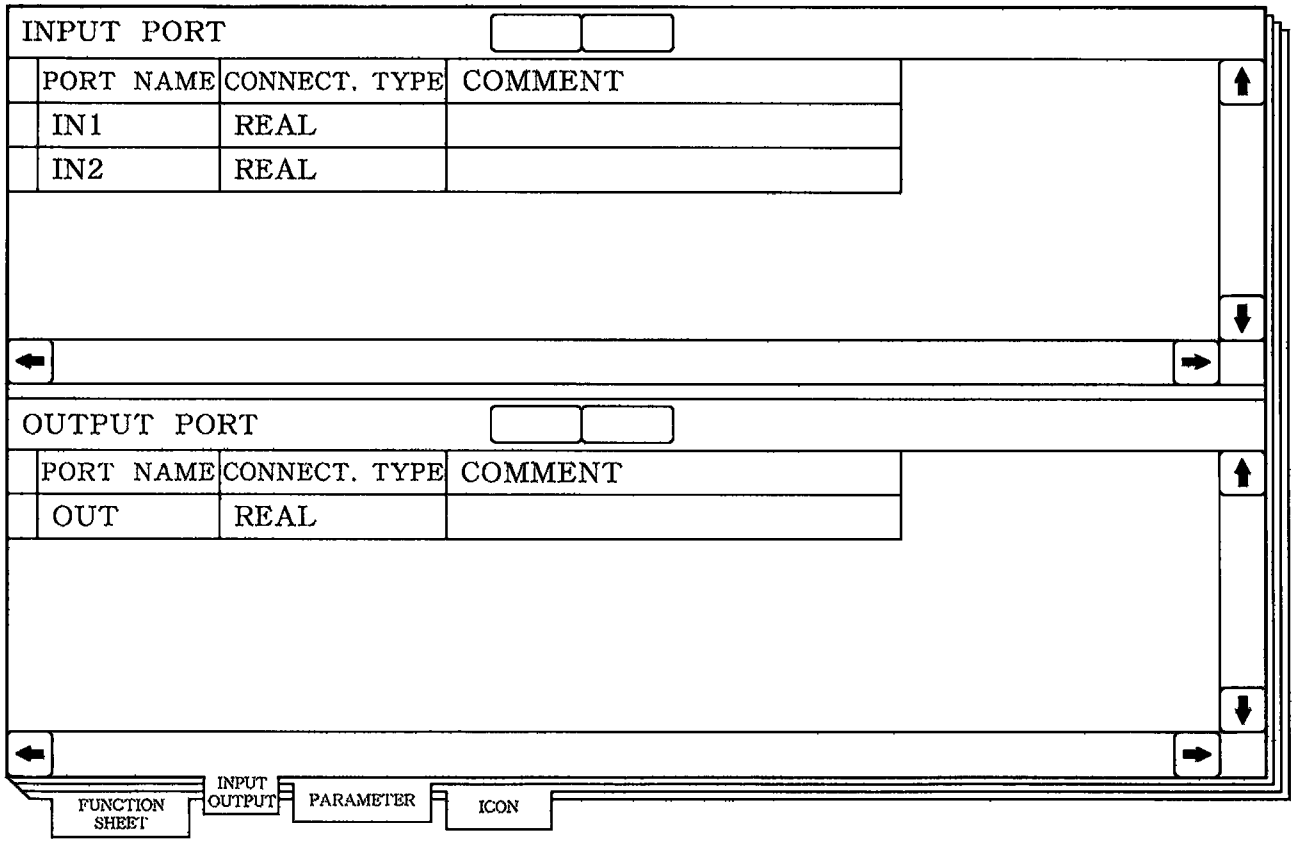

INPUT PORT
PORT NAME
CONNECT. TYPE
COMMENT
IN1
REAL
IN2
REAL
OUTPUT PORT
PORT NAME
CONNECT. TYPE
COMMENT
OUT
REAL
FUNCTION SHEET
INPUT OUTPUT
PARAMETER
ICON

FIG. 12

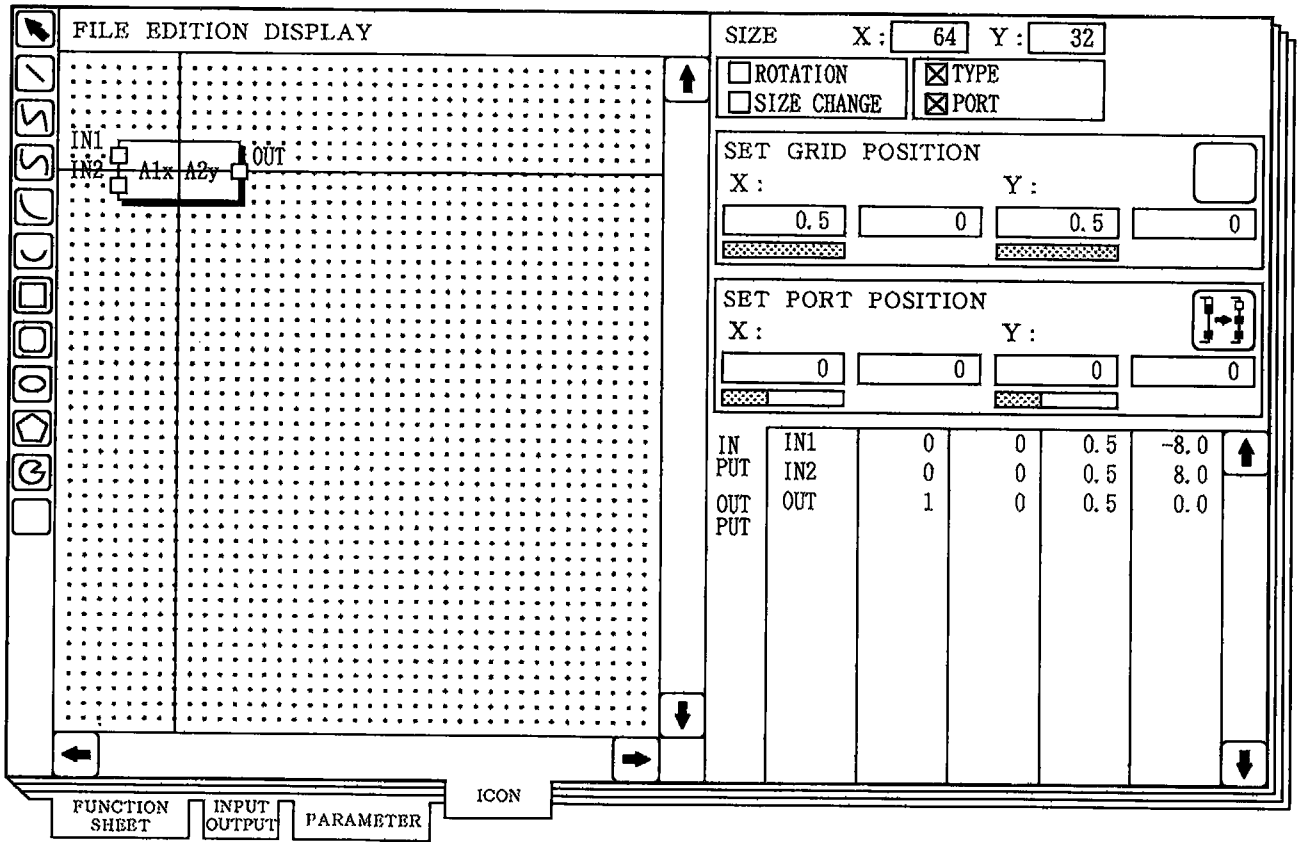

FILE EDITION DISPLAY
IN1
IN2
A1x
A2y
OUT
SIZE X: 64 Y: 32
ROTATION
SIZE CHANGE
TYPE
PORT
SET GRID POSITION
X:
Y:
0.5
0
0.5
0
SET PORT POSITION
X:
Y:
0
0
0
0
IN PUT
OUT PUT
IN1 0 0 0.5 -8.0
IN2 0 0 0.5 8.0
OUT 1 0 0.5 0.0
FUNCTION SHEET
INPUT OUTPUT
PARAMETER
ICON

FIG. 13

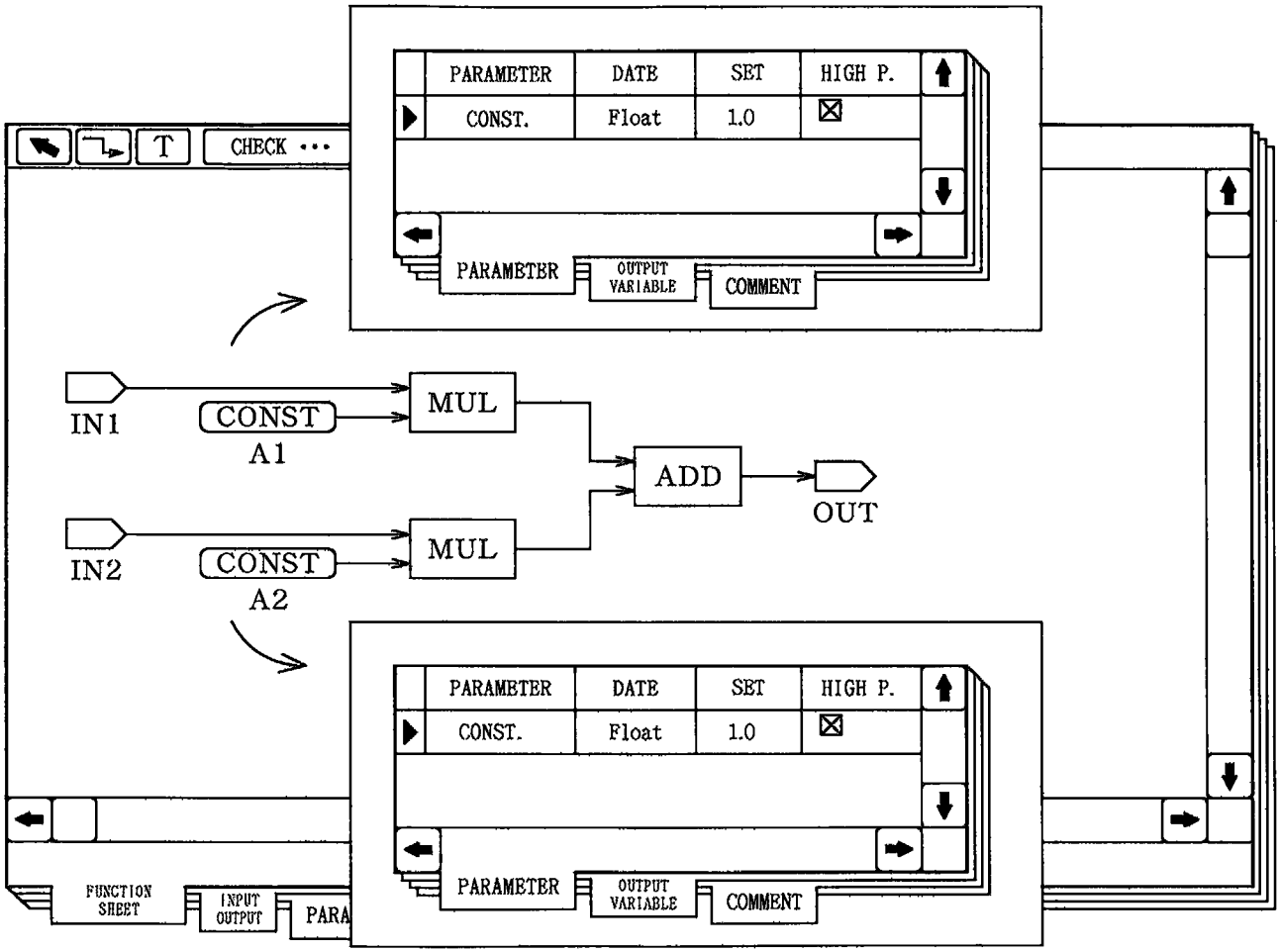

PARAMETER
DATE
SET
HIGH P.
CONST.
Float
1.0
PARAMETER
OUTPUT VARIABLE
COMMENT
CHECK
IN1
CONST
A1
MUL
ADD
OUT
IN2
CONST
A2
MUL
PARAMETER
DATE
SET
HIGH P.
CONST.
Float
1.0
PARAMETER
OUTPUT VARIABLE
COMMENT
FUNCTION SHEET
INPUT OUTPUT
PARA

FIG. 14

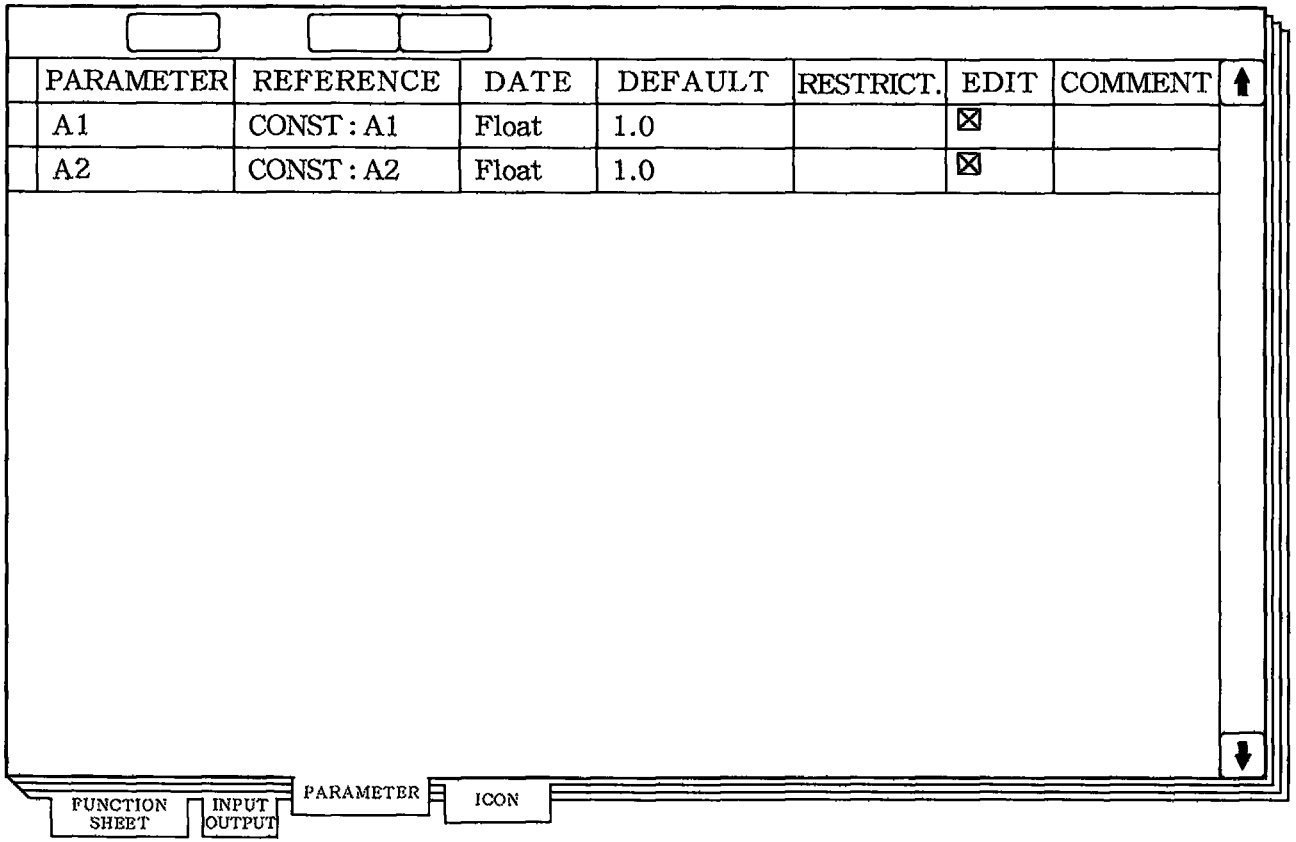

PARAMETER
REFERENCE
DATE
DEFAULT
RESTRICT.
EDIT
COMMENT
A1
CONST : A1
Float
1.0
A2
CONST : A2
Float
1.0
FUNCTION SHEET
INPUT OUTPUT
PARAMETER
ICON

FIG. 15

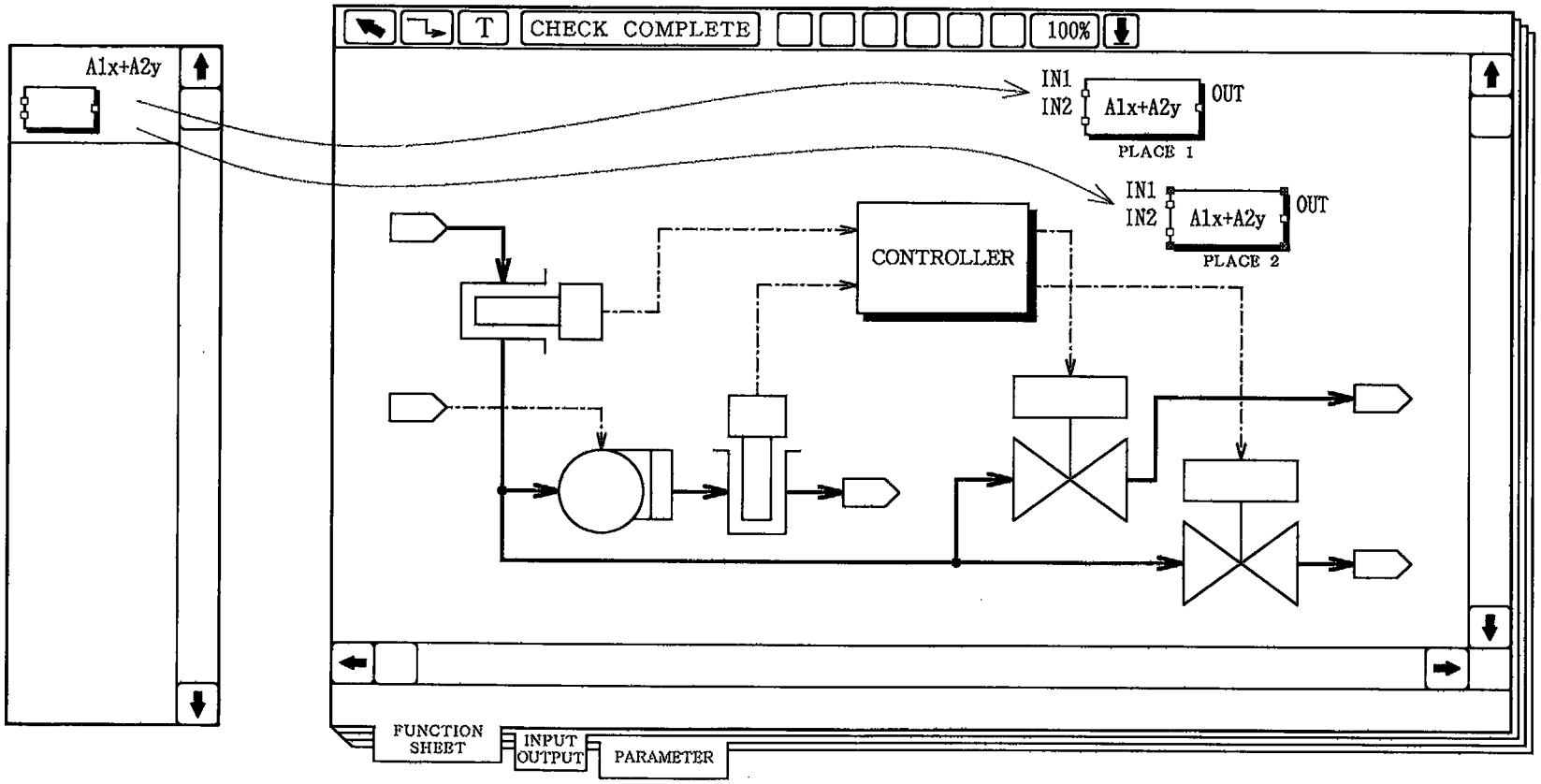
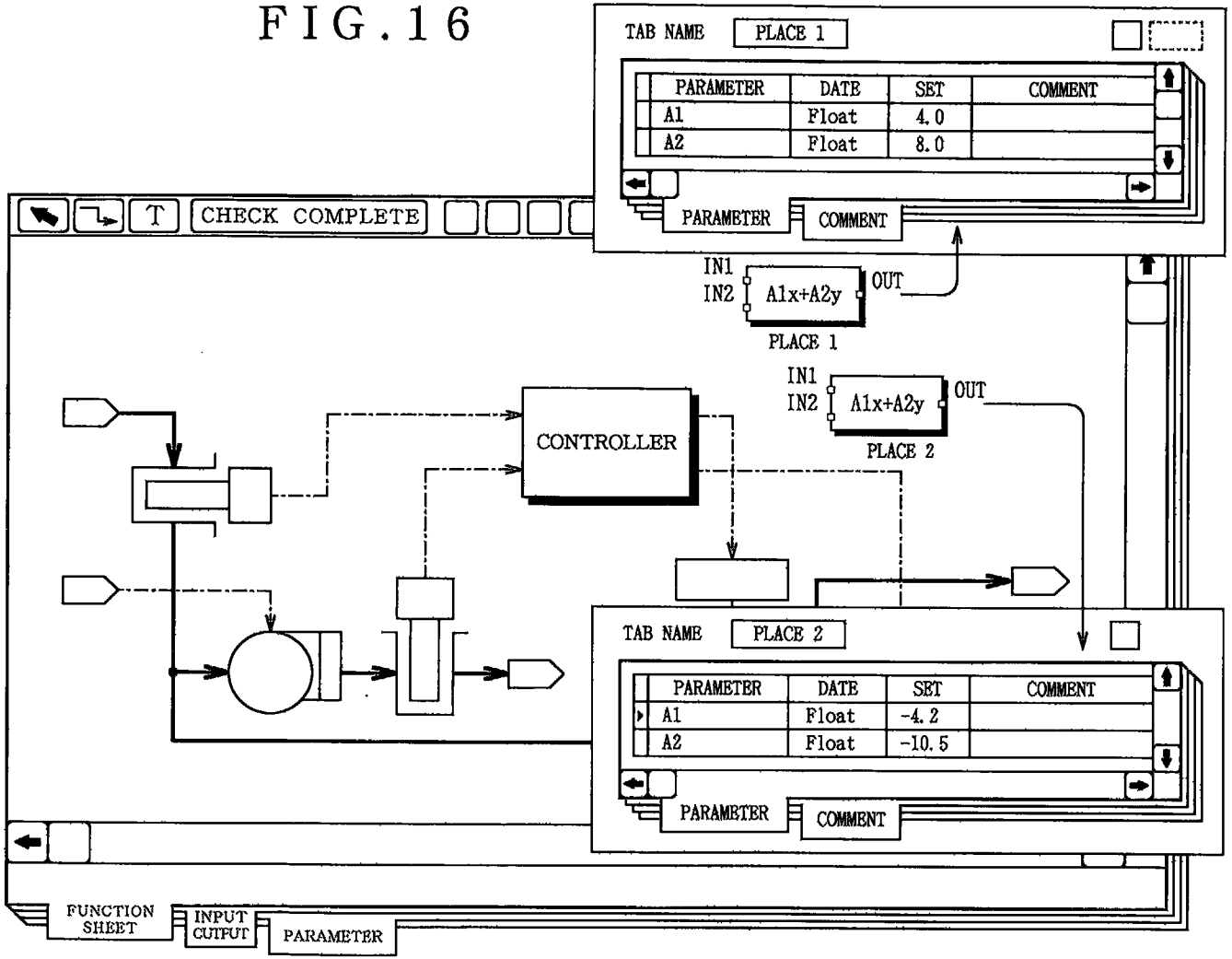

A1x+A2y
T
CHECK COMPLETE
100%
IN1
IN2
A1x+A2y
OUT
PLACE 1
IN1
IN2
A1x+A2y
OUT
PLACE 2
CONTROLLER
FUNCTION SHEET
INPUT OUTPUT
PARAMETER

FIG. 16


TAB NAME
PLACE 1
PARAMETER
DATE
SET
COMMENT
A1
Float
4.0
A2
Float
8.0
PARAMETER
COMMENT
CHECK COMPLETE
IN1
IN2
A1x+A2y
OUT
PLACE 1
IN1
IN2
A1x+A2y
OUT
PLACE 2
CONTROLLER
TAB NAME
PLACE 2
PARAMETER
DATE
SET
COMMENT
A1
Float
-4.2
A2
Float
-10.5
PARAMETER
COMMENT
FUNCTION SHEET
INPUT OUTPUT
PARAMETER