PROGRAMMABLE CONTROLLER

The present invention is directed generally to a programmable controller, into which a sequence control program is incorporated, for inputting and outputting information to a variety of local intelligent appliances in an FA (Factory Automation) field for efficiently controlling factory production lines or in a PA (Process Automation) field for controlling multiple industrial processes.

The present invention is directed more particularly to an improvement of a change-over system associated with a 1-bit processor for executing the sequence control program and an ordinary process. Concomitantly, there are ameliorated a function for a basic process arbitrarily set by the user, a control function over a group of I/O cards for transferring signals to the local appliances and receiving the signals therefrom and a programming workability of the sequence control program set inside.

In general, a CPU board in the programmable controller is equipped with a typical general-purpose processor (e.g., a 16 bit microprocessor 68000 or the like) serving as a first processor and a processor (e.g., a 68000 family or the like) dedicated to execute only an arithmetic operation of numeric values or logic which serves as a second processor. These processors operate while being changed over in accordance with a sequence control program stored in a program memory.

An example of a module, wherein the first and second processors alternately operate to read from the single program memory is described below in relation to Fig. 1.

This type of prior art programmable controller utilises a combination of the first and second processors, resulting in such a troublesome situation in terms of design that the machine language commands stored

in the program memory have to be allocated beforehand to the first and second processors respectively at a stage of system design.

On the other hand, recently oft-utilised commands are interpreter type commands (interpretation program commands) designed independently of the machine language commands intrinsic to the first processor, these interpreter type commands tending to be stored in the program memory.

An apparatus making use of the interpreter type commands is capable of freely setting commands executed by the second processor in addition to the commands executed by the first processor. There is no necessity for allocating the commands to the first and second processors respectively at a stage of designing the first processor. The first or second processor is specified in conformity with contents of the commands read from the program memory, and it is therefore possible to increase a practical efficiency and expand the commands to be executed by the second processor.

In the case of employing the machine language commands and the interpreter type commands, however, the first processor, when the second processor operates, sequentially reads the program commands from the program memory and supplies these commands to the second processor, which in turn leads to a problem of causing a time-consuming transfer of the program.

Where the interpreter commands are used, the first processor has to interpret the readout commands and determine a processor for execution. Especially in an apparatus where the first and second processors are frequently changed over, an excessive period is created.

It is a primary object of the present invention, which obviates the foregoing problems, to provide a programmable controller capable of facilitating an operation of changing over the first and second

processors and speeding up processes as a whole.

According to the present invention, a programmable controller comprises: a first processor; a second processor; a program memory system for storing interpreter type program commands and basic commands; a first information bus through which said first processor is connected to said program memory system; and a second information bus through which said second processor is connected to said first processor; whereby said first processor reads a program command and judges whether said program command be executed by said first processor or said second processor; when said first processor judges that said program command be executed by said first processor, said first processor itself operates, said first processor reads said program command from said program memory preparatory to execution of said program command and at the same time issues a pseudo command to said second processor; and when said first processor judges that said program command be executed by said second processor, said first processor supplies said program command to said second processor in place of said pseudo command.

In order that the invention may be clearly understood and readily carried into effect examples thereof will now be described in relation to an example of the prior art with reference to the accompanying drawings, in which:

Fig. 1 is a block diagram illustrating a prior art programmable controller;

Fig. 2 is a block diagram illustrating an embodiment of a programmable controller of the invention;

Figs. 3(a) and 3(b) are diagrams each showing operations of the apparatus depicted in Fig. 2;

Fig. 4 is a block diagram illustrating an apparatus, wherein a construction of the embodiment of Fig. 2 is partly modified;

Figs. 5(a) and 5(b) are diagrams each showing operations of the apparatus depicted in Fig. 4;

In the prior art module depicted in Fig. 1, a first processor CPU1 conceived as a typical general-purpose processor (e.g., the 16-bit microprocessor 68000) and a second processor BPU2 defined as a processor (e.g., the 68000 family) dedicated to execute the arithmetic operation of numeric values or logic are connected via an information bus b to a program memory 3. Note that the second processor BPU2 is in some cases referred to as a 1-bit processor.

The program memory 3 stores machine language commands as program commands for the processor CPU1. The processor CPU1 executes processing by sequentially reading the machine language commands. The processor CPU1, when the readout commands are to be executed by the processor BPU2, gives instructions for an operation and a data preparation to the processor BPU2. Subsequently, the commands read from the program memory 3 are sequentially issued to the processor BPU2 with a view to imparting a right of control execution.

In Fig. 2, there is shown an embodiment of a programmable controller of the invention.

In this embodiment, CPU1 is connected via a first information bus b1 to BPU2 which is in turn connected via a second information bus b2 to the program memory 3.

The processor BPU2 incorporates a program command take-in circuit 21 connected to the information bus b2 on the side of the program memory 3, a program command readout circuit 22 connected to the information bus b1 on the side of CPU1 and further a pseudo command generating circuit 23 for holding a pseudo command representing "no execution" with respect to CPU1. The program command readout circuit 22 includes a change-over circuit 24 for selecting the program command take-in circuit 21 or the pseudo command generating circuit 23.

In this embodiment, the program memory 3 stores interpreter type commands and the BPU2 basic commands as well.

The operations of the thus constructed apparatus of the invention will be explained with reference to Figs. 3(a) and 3(b).

Fig. 3(a) illustrates a case where CPU1 executes the commands.

The processor BPU2 judges that the command 1 read at an address Y1 is catered for CPU1 and changes over the change-over circuit 24 to the program command take-in circuit 21 within a dedicated processor 2. The readout command 1 is imparted from the program command readout circuit 22 via BPU2 to the processor CPU1 which in turn starts operating.

Note that the readout addresses with respect to the program memory 3 are generated from BPU2 even during operation of CPU1.

Turning to Fig. 3(b) there is shown a case where BPU2 executes the commands.

The processor BPU2 judges that the command 2 read at an address Y2 is to be processed by itself, whereby the pseudo command circuit 23 is connected to the program command readout circuit 22 with the aid of the change-over circuit 24. With this arrangement, BPU2 reads the command from the program memory 3 in accordance with the readout address generated by itself and then executes the command, while CPU1 performs nothing in conformity with a pseudo command N representing "no execution" which is given from BPU2.

In this manner, the commands read from the program memory 3 are surely imparted to CPU1 or BPU2, thereby smoothly effecting change-over between CPU1 and BPU2 without causing futility in time.

In the example shown in Fig. 2, the processor BPU2 is middled between CPU1 and the program memory 3.

However, another arrangement equivalent thereto is that CPU1 may be middled therebetween by providing circuitry in CPU1, which corresponds to the program command take-in circuit 21, program command readout circuit 22, pseudo command generating circuit 23 and change-over circuit 24.

In this example, the processor interposed therebetween acts to generate addresses with respect to the program memory 3. Hence, even after changing over the processor which is to execute the command, a situation of advancement of the program is constantly grasped by the middled processor, and it is therefore possible to omit a period for transmitting and receiving the addresses with respect to the program memory 3.

Turning now to Fig. 4, there is given an arrangement in which the addresses generated on the side of CPU1 can be sent to the program memory 3 in the embodiment of Fig. 2.

In this example, the processor BPU2 incorporates an address reading circuit 25 for taking in addresses generated by CPU1 toward the information bus b1, an address transferring circuit 26 for imparting the readout addresses of the program memory 3 to the information bus b2 and a change-over circuit 27. The change-over circuit 27 serves to connect the address transferring circuit 26 to the address reading circuit 25 or to an address generating circuit provided in an interior of the dedicated processor 2.

Operations of this embodiment will hereinafter be described in conjunction with Figs. 5(a) and 5(b).

Fig. 5(a) illustrates a case where CPU1 executes the commands. The change-over circuit 24 incorporated in BPU2 is connected to the program command take-in circuit 21, while the change-over circuit 27 is connected to the address reading circuit 25. An address Z1 generated in CPU1 is transferred via BPU2 to the program memory 3, and the command 1 corresponding thereto passes through BPU2

and is then given to CPU1.

Fig. 5(b) shows a case where BPU2 executes the commands. The change-over circuit 24 incorporated in BPU2 is connected to the pseudo command generating circuit 23, while the change-over circuit 27 is connected to the address generating circuit 28. The processor BPU2 reads and executes the command 2 in accordance with an address Y3 generated inside, whereas CPU1 executes nothing after reading a pseudo command N from BPU2.

The arrangement exemplified in Fig. 4 makes CPU1 accessible to an arbitrary position of the program memory 3 and is also advantageous when executing a jump command or the like.

In accordance with the embodiment shown in Fig. 2, or 4, the processors CPU1 and BPU2 thus operate alternately in conformity with the commands read from the program memory 3. Especially BPU2 reads the commands directly from the program memory 3, and hence high-sped processing is practicable, correspondingly.

Since the basic commands may be defined as those intrinsic to BPU2, there is no necessity for interpretation on the occasion of execution by BPU2, and it is therefore possible to omit this period.

As discussed above, the programmable controller of the invention is capable of improving a velocity of sequence control processing and attaining an easy-to redesign system when being modified. The programmable controller exhibiting a high processing efficiency can thus be actualised.

Although the illustrative embodiments of the present invention have been described in detail with reference to the accompanying drawings, it is to be understood that the present invention is not limited to those precise embodients. Various changes or modifications may be effected therein by one skilled in the art without departing from the scope of the following

claims.

Patent Application GB 2244828 has been divided from the present application and according to that application a programmable controller comprises: a group of I/O boards for transferring and receiving multiple information with respect to a control object; and a processor board for imparting a control signal to said control object via said I/O board group, said processor board including: a processor for controlling the whole and executing a part of commands in a sequence control program which is made by use of sequence basic commands and sequence application commands and a BASIC program for performing general-purpose arithmetic processing, information processing or a control operation by starting said sequence control program and making an end instruction; a 1-bit processor, connected directly to a program memory into which said sequence control program is stored, for reading commands sequentially from said program memory, for executing said commands if said read out commands are commands to be executed by said 1-bit processor and for imparting said commands to said processor if said readout commands are sequence application commands to be executed by said processor; a data memory for temporarily storing data; a fixed memory for storing a self-diagnostic program; a communication interface participating in a communicating operation with a host computer; an I/O interface for joining said control object to an I/O bus through which I/O boards for transferring and receiving multiple information are connected; and an internal bus for mutually connecting said processor, said 1-bit processor, said data memory, said fixed memory, said communication interface and said I/O interface.

* C L M S *

CLAIMS:

1. A programmable controller comprising: a first processor; a second processor; a program memory system for storing interpreter type program commands and basic commands; a first information bus through which said first processor is connected to said program memory system; and a second information bus through which said second processor is connected to said first processor; whereby said first processor reads a program command and judges whether said program command be executed by said first processor or said second processor; when said first processor judges that said program command be executed by said first processor, said first processor itself operates, said first processor reads said program command from said program memory preparatory to execution of said program command and at the same time issues a pseudo command to said second processor; and when said first processor judges that said program command be executed by said second processor, said first processor supplies said program command to said second processor in place of said pseudo command.

2. A programmable controller substantially as hereinbefore described with reference to Figs. 2 to 5(b) of the accompanying drawings.