Design, Implementation, and Testing Of a Control System for Heat Treatment Furnaces Crane System Based On a Single Chip Microcontroller

Miss. Ameera Ali Salman Al-Qaissi*

Received on: 10/10 /2003
Accepted on: 1 /11/ 2004

Abstract

Design, implementation, testing, and installation of a control system for a Heat Treatment furnaces crane system are carried out. A simple structure for the control system is defined as to meet the hardware functional requirements of sensing the input variables, commanding actuators. The basic hardware parts of the control system are based on the single chip microcontroller unit and the associated logic circuit. Micro controllers are found in small, minimum-component designs performing control-oriented activities; these designs were often implemented in the past, using dozens or even hundreds of digital ICs. So micro controllers are suited to "control" of I/O devices in designs requiring minimum components count. The basic software functions are turning the control system hardware into virtual high-level machine. It includes basic driver routines (DI/O), control algorithm and management software routines.

الخلاصة

تم القيام بتصميم، وتنفيذ، وفحص، ونصب منظومة سيطرة لنظام رافعة الافران المعالجة للحرارة. وتم تعـــريف تركيـــب بسيط لمنظومة السيطرة لكي يلبي المتطلبات الوظيفية المادية من متحسس لمتغيرات الادخال والســيطرة على المنفذات. وقد صممت الاجزاء المادية الاساسية لمنظومة السيطرة بالاعتماد على وحدة المعالج الدقــيق ذي الشــريحة الواحدة والدوائر المتعلقة بها. وجدت المسيطرات الدقيقة في التصاميم ذات المواد القليلة والصــغيرة لتنفــيذ فعاليات السيطرة، في حين ان هذه التصاميم كانت تنفذ في الماضي باستعمال مئات الشرائح الرقمية. لذا، فان المسيطرات الدقيقة ملائمة "للسيطرة" في التصاميم ذات الادخال والاخراج التي تحتاج الى عدد قلــيل من المواد او الشرائح الرقمية. ان الوظائف الاساسية للبرامجيات هي التي تدير الاجزاء المادية للمنظومة الـــى الة بمستوى عملي عالي. وتتضمن البرامج السواقة الثابتة (الادخال والاخراج الرقمي)، وبرامج السيطرة والادارة.

*Control & Systems Engineering Department, University Of Technology, Baghdad, Iraq.

1. Process system description

The main point of the process is to obtain steel pieces with a certain properties by heating the materials to a certain temperature degree and other conditions like cooling using water, air or oil.

The process system structure can be divided into four regions as shown in Fig (1): the first region (furnace region) has six heat treatment furnaces, which are used for heating materials (charge) for a certain temperature degree according to certain tables. The second region has two quenching baths used for cooling the charge for a certain period using either oil or water. According to the type of the material and these baths consisting of tables that can be lifted or lowered to load or empty the charge. The third region is the charge tables; it consists of eight tables which are arranged as two horizontal lines, i.e. each four tables in a line.

Finally, the fourth region is the crane region which acts as a big stroke that can be moved in three dimensions:

- Higher and lower;
- Forward and backward;
- Right and left.

This stroke can be moved along a railway line. Fig (1) shows the description of this process. The detailed and global system design is shown in FIG. (2); which represents the control system.

2. System Design

To produce a flexible system, careful consideration of the main points of the system specification is required:

1. Low cost, and high (maximum) functionality are desirable for the smallest board area possible. systems and microcontrollers with many features are selected to provide this.
2. Simplicity: by using standard large scale memory, multiple function devices can be obtained and the overall circuit complexity can be reduced.
3. Flexibility: high modular distribution of functions between cards allows the user to configure the system for his application with the minimum of redundancy and the lowest chip count.

The detailed and global system design is shown in Fig. (2), which represents the control system of the material and the module that has an AT89C51 as a central processing unit. Both the input and output channels are serial and 12v ranged. There are isolated digital inputs and digital outputs. The input and output slots have a noteworthy point; they can be configured either as input slot or and output slot because each slot of the I/O card has an 8-bits for card address, 3-bit for bit selection, one bit for serial input, and one bit for output; so that each slot can be configured as input or output card (see Fig 3).

3. Hardware description

The system is based on a microcontroller AT8951 card plugged on a rack representing the motherboard card of the control system. Twelve of 8-bits digital input and eight of 8-bits output cards are also put on specified slots of the control system case; a detailed description below will explain each section of the hardware control system.

3.1Microcontroller Description

The microcontroller-based control system is designed and implemented using a single chip microcontroller Atmel 89C51. This chip is also compatible with Intel's 8051-microcontroller family. So it's suitable processing power and multiple on-chip resource that can minimize and simplify the design in a great deal. An important feature of microcontrollers is the built in interrupt system. As control oriented devices, microcontrollers are often called upon to response to external interrupts in real time. They must perform fast context switching, suspending one process while executing another in response to an "event". This microcontroller has a multiplexed address-data bus, which requires an octal latch to latch the lower address byte and when address latch is enabled (ALE) signal goes low. This permits the same port (P0) to accept: instruction from the EPROM data lines when the Program Store Enable (PSEN*) signal enables the EPROM output. The CPU (AT89C51) runs on a 12 MHz clock, which gives a basic instruction cycle of 1 microsecond. The external access signal (EA*) of the CPU is forced to low in order to execute an instruction from external EPROM, but when it wants the CPU to execute an instruction from the internal 4Kbyte EEPROM, the (EA*) signal will be connected to Vcc. The (RD*) and (WR*) signals are used to address the data space, while the (PSEN*) signal is used to address the program space. In this way the address, data and control buses are generated to be used to tie up other controller components of memories, PPI ...etc. The main features of the microcontroller can be summarized below:

- 4Kbytes flash EPORM.
- 128 Bytes RAM.
- Four 8-bits (32 bits) I/O ports.
- Boolean processor.
- 210 bit –addressable locations.
- 4μs multiply/divide.
- Two 16-bit timers.
- Serial interface.
- 64Kbytes external code memory space.
- 64Kbytes external data memory space.

3.2 System Clock and Reset Circuit

The process controller system timing is provided with a crystal oscillator module, which has a frequency of 12 MHz. This clock frequency is applied directly through the clock pins (19 & 18) X1 and X2.

The AT89C51 requires that an external reset must be applied to allow stabilization of the on-chip circuitry. Push-button reset circuit is employed to reset the system automatically to guarantee that the CPU behaves in a predictable fashion. In general, the circuit consists of RC circuit (R1 and C1) where the RC would give a time constant that determines both the power on reset signal and the manually reset that is generated using a switch.

3.3 Memory Section

In general, the memory section of the microcontroller consists of three types of memories, which are Stack-RAM, RAM (data memory), and EPROM (program memory).

The stack-RAM includes 128bytes of the internal RAM that is residing in the AT89C51 package. It would be so-called "stack-RAM" because the AT89C51 stack is built automatically in this RAM when using the instruction "PUSH" also to store the program pointer, when using the instruction "CALL", or when an interrupt occurs. It is addressed using one byte address; the space reserved for the internal RAM is physically separated from the space that the external data memory may use. The lowest 8 Bytes in the RAM are selected to be the working registers of the CPU, and can be accessed by a 3-bit address in the same byte as the opcode of an instruction. The stack pointer is 8-bit wide. The stack can be resided anywhere in the on-chip RAM. The stack pointer is initialized to 07h when AT89C51 is reset, while executing a "PUSH" or a "CALL" instruction increments the stack pointer before data is stored, so the stack would begin at location 08h. The microcontroller contains a 210 bit addressable locations, of which 128 are at byte addresses, 20H through 2FH, and the rest are in the special function registers. The idea of individually accessing bits through software is a powerful feature of most microcontrollers. Bits can be set, cleared, ANDed, ORed, etc., with a single instruction.

Furthermore, the microcontroller I/O ports are bit–addressable, simplifying the software interface to a single–bit inputs and outputs. So there are (8bits/ byte* 16bytes=128bits); these addresses are accessed as a byte or as bits, depending on the instruction used.

3.4 Memory Mapping

In this design, the internal data memory of the microcontroller (128 byte) was sufficient to store the digital inputs signals and this will depend on the number of the connected input/output cards.

For the output card, the address 2CH to 2FH is also reserved to be an output data as shown in table 1.

Card type	Size(byte)	Memory location
Input card	12 byte*	20H-2BH
Output card	4 byte*	2CH-2FH

(Table 1)

Some of the addressable bits were used to indicate as an alarm a signal which is also output to the field as a serial data.

For the digital input, a single bit of port2 (P2.6) was used to read the serial input data and the serial output was implemented using also a single bit of port2 (P2.7). All the addresses used in this design were bit addressing because all the input and output actions are serial except the selected card number and the specified bit number on the selected card that must be 8-bits or less parallel data.

Table 2 shows the specified bits used to address each function of the card.

Function	Addressable bits
Digital input	P2.6
Digital output	P2.7
Card address	P0.0-P0-7
Selected Bit	P1.0-P1-3

(Table 2)

3.5 Digital Input Section

These lines are divided into two types:

- The first type is an isolated serial data input from the plant by means of Opto-coupler IC 4N26. This is a good method for isolating digital source signals from the microcomputer system. It consists of a light emitting diode (LED) powered from the plant (sensor) side and phototransistor acting like a switch that closes when the (LED) is on. Since there is no conductive connection between the grounds of two sides of the isolator, the common mode signal is eliminated. In addition, the isolation of the input/output points in industrial environment is very important. So this circuit enables the controller to accept different signals from various kinds of digital sensors.

In this process, the number of the digital inputs cards is twelve and they are put in specified slots of the control system case. After reading the serial data input, the reading values are stored in the addressable bit area of the microcontroller so that it can be read as a bit or as a byte.

- The second type of digital input is a thumb switch that is represented as four

switches used testing or configuring a certain bit in a certain card.
More details can be seen in Fig (4) that shows the internal structure of the input card.

3.6 Digital Output Section

The digital output data can be also divided into two types:

- The first type is one serial data output; it is implemented as a relay type single pole double through (SPDT) which may give a flexible range to connect different kinds of actuators to the controller. Open collector buffers (inverted 7406 and no inverted 7407) were used to drive the relays. In addition, a Darlington transistor driver uses arrays to cover a range of 500mA or 1.5A output current.
- The second type of the digital output is also implemented as a relay of type SPDT, so eight bits of the microcontroller ports were connected through a driver IC to eight relays to enable the selection of card select address whether it was input card or output card. In addition, three bits of the microcontroller were connected through another driver IC to three relays of the same type (SPDT) to enable the selection of a certain bit in a certain card in order to test or configure it to a specified state. Fig (5) shows the internal structure of the output card.

4. Software description

The main program is entered after applying reset pulse to the AT89C51, and the complete microcontroller software on the CPU card contains a number of program modules:

- INIT: for initializing program variables immediately after the system reset, these variables represent memory location points to a certain input or output data. Several variables are to be initialized before the infinite loop in the main program begins. First initialize various I/O ports and timers by a subroutine named INIT (resetting or clearing the output variables at the start of the main program).
- The microcontroller machine monitor is able to give or receive instructions and data. A memory management routines and testing functions are also included in the final system.
- Control algorithm: this code module is located in the internal program memory (4Kbyte flash EPROM) of the microcontroller and it is executed as a stand alone program unit. The algorithm employed type is a sequential control; a sequence of events executed sequentially to produce a certain action such as moving stroke left or any other actions. Below are some samples of the events accrued to provide an action.

Types of movements for Crane:

Outputs

Contactors (serial outputs):

1. Stroke lift.
2. Stroke lower.
3. Hydraulically pump on.
4. Machine travel forward slow.
5. Machine travel forward fast.
6. Quenching time bath1.
7. Quenching time bath2.
8. Trolley travel on.
9. Trolley travel forward slow.
10. Trolley travel forward fast.
11. Trolley travel return slow.
12. Trolley travel return fast.
13. Emergency Horn.

Lamps:

14. Furnace1 busy.
15. Furnace2 busy.
16. Furnace3 busy.
17. Furnace4 busy.
18. Furnace5 busy.

19. Furnace6 busy.
20. Bath1 busy.
21. Bath1 busy.

- To serialize this process, it is required to design a subroutine to put each bit of a specified input at the addressable bit area. So a task list for such as routine is:

```
1. Count =8.
2. While count >0 do:
    a) Shift data left into carry.
    b) Decrement counter by one.
    c) Go to 2.
3. Read serial data and store it in the
   addressable bit area.
```

Below is a list of the types of inputs signals:

Inputs (serial bits):

1.Machine in front of furnace1.
2.Machine in front of furnace2.
3.Machine in front of furnace3.
4.Machine in front of furnace4.
5.Machine end left side.
6.Machine end right side.
7.Succinct right side.
8.Succinct left side.

Stroke height:

1.1050 deviation furnaces lifted.
2. 950 entering furnaces.
3. 600 lifted deviation.
4. 400 lowered deviation.
5. Emergency off.
6. Circulation pumps bath1 on.
7. Circulation pumps bath2 on.
8. Machine travel right side slow.
9. Machine travel left side slow.
10. Stroke lift.
11. Stroke lower.
12. Lamps check.
13. Bath1 occupied/free.
14. Bath1 occupied/free.
15. Charge in Bath1.
16. Charge in Bath2.

- The output subroutine gets a certain output, which was produced according to logical functions (conditions) that are stored in the addressable bit area, and sends the final output condition to the field to move or open a gate or a stroke. So a task list for such as routine is:

```
1. Count =8.
2. While count >0 do:
    a) Shift data left into carry.
    b) Decrement counter by one.
    c) Go to 2.
3. Send output serial data to plant.
```

The software programs have been written in low-level language in the side of the control system.

5. Software Development Tools

The computer was used to aid in translating code from source code to object code (or machine code) and loading this into memory through a hexadecimal loader that will translate this code into binary and put the code in designated memory locations. This loader might be part of the software microcontroller programmer.

In the following, developing a certain program from the IBM personal computer to micro controller will be explained in detail:

- A text editor for creating and modifying a text file can be performed by using the editor of the Windows "Notepad.exe". So, when the user wants to create a new file, he/she should specify a name for his/her new program with the extension .asm. On the other hand, when the user wants to open an existing file he can edit it.
- The assembler of the MCS-51 family is used to translate the MCS-51 assembly language modules into object code modules, and it is named "SXA51.exe" that will convert the source program (.Asm) into object file. It lists file and error diagnostics, provides software support for many addressing and data allocation capabilities, and provides

(.Lst, .Hex). This assembler will be run under MS-DOS environment.

- After writing or opening an application program and assembling it to a file in Intel hex format, this file must be converted to a file in binary format, which can be achieved using a special function named "hex2bin.exe".
- The last step; the interface between personal computer and the controller should be provided through parallel port cable of the programmer of 8951 devices.

6. Conclusions

Through this work, a cheap, versatile and easy control system module has been designed and implemented. The design is achieved through the use of AT89C51 microcontroller to work as the CPU of the module. This chip is used because of its availability, being tested, and its aid in reducing the overall components count. It is run at a speed up to 24MHz, and internally consists of 4KByte Flash EPROM that makes it ideal as a development tool. So this CPU can handle data acquisition and analysis between the controller and processes under control. The microcontroller card can also change its program through using development tools, like programmer, to develop any application or test program to the microcontroller module. In this work, a decision has been made to design the controller module using single board because no extra components are required, and all components are not taking up large space and power. In addition, the design of a single board has been also made for simple industrial use. It has been decided to include an expansion bus to provide a sufficient choice of I/O devices on the PCB to satisfy the most applications.

Optical coupling devices completely separate the micro controller's signals from the factory signals, and because each module has a separate ground wire to the factory device it is connected to, it is completely isolated from other grounds. When several I/O devices share a common ground, ground loop currents can induce signal noise. Maintaining isolated grounds may be an important consideration in retrofitting older equipment.

The software of control system was written in the low level language (assembly language of AT89C51) and it was located in the internal 4Kbytes flash EPROM (program memory).

The main program of the controller is designed as a single-user system, which contains two programs. The first is the foreground program, which interacts with the terminal and runs as long as it is able. The second is the background program, which is assigned to the processor whenever the foreground program is unable to proceed.

The structure of the main program was represented as a collection of sub-routines:

First, initialize routine, then read serial input data and store it in the internal RAM of the microcontroller.

Second, perform the control algorithm and store the results in the internal memory too.

Third, finally drive out the serial data to a selected output card.

The programs were written using assembly language of the AT8951 microcontroller.

There are two layers of the software: the first layer represents the application standard routine that includes digital input, output routines and the control software required for application. As this layer is located near the hardware, it must be written in assembly language and translated into object code to stand for the microcontroller. The second layer represents all the routines that are used to create any application program without taking into account the hardware structure. It includes the operating system of the controller (kernel software), which is usually a fixed layer. The main program defines various constants, initializes ports and devices, if needed, and enters an infinite loop. The main program is entered after applying RESET pulse to the AT89C51. Thus, before calling INIT subroutine, it initializes the stack pointer, recall that the stack growing from the high address to the low address. Then initiate the I/O cards and read the states of each point in the selected card and store this data in the internal memory of the microcontoroller. After applying the specified fiunction based on the control algorithm, out the result to the selected bit in a certain card in order to perform the wanted movement of the crane.

References

1-Mohammed K. Refai: "Microprocessor Based Digital Controller for DC Motor Speed Control", Microprocessor and Microsystems Vol.1, No.10, December 1986.

2- Trevor Rolls, Ashley Willis & Roy Gwinn: "Design Concepts for A Microprocessor Based Temperature Control", Microprocessor and Microsystems, Vol. 6, No.5, June 1982.

3- Jim Tebbs: "Tracking Over-Temperature Protection Devices", Application notes, Eurotherm controls limited, 1998.

4- Ian Milne: "Stream Valve Rapid Shut-Off Control", Application notes, Eurotherm Controls Limited, 1998.

5- Paul F. Lister: "Single chip Microcomputers", Granada, 1994.

6- Texas Inc.: The TTL Data Book for Designs Engineers, Texas instruments, 1989.

7- Stojic MR: "Design Of A microprocessor-Based Digital System for DC Motor Speed Control ", IEEE Trans. Industry, Vol.IE-31, No.3, pp.243-248, August 1984.

8- Michael Babb: "Using a Personal Computer for High-Speed, Real-Time Machine Control", Control Engineer, Vol.35, No.11, pp 68-72, November 1988.

9- I.Scott Mackenzie: "the 8051 microcontroller", Prentice Hall, 1999.

10- Michael Babb:" Single Point I/O Modules Have an "Optimistic" future", Control Engineer, Vol.37, No.4, pp 66-68, March 1990.

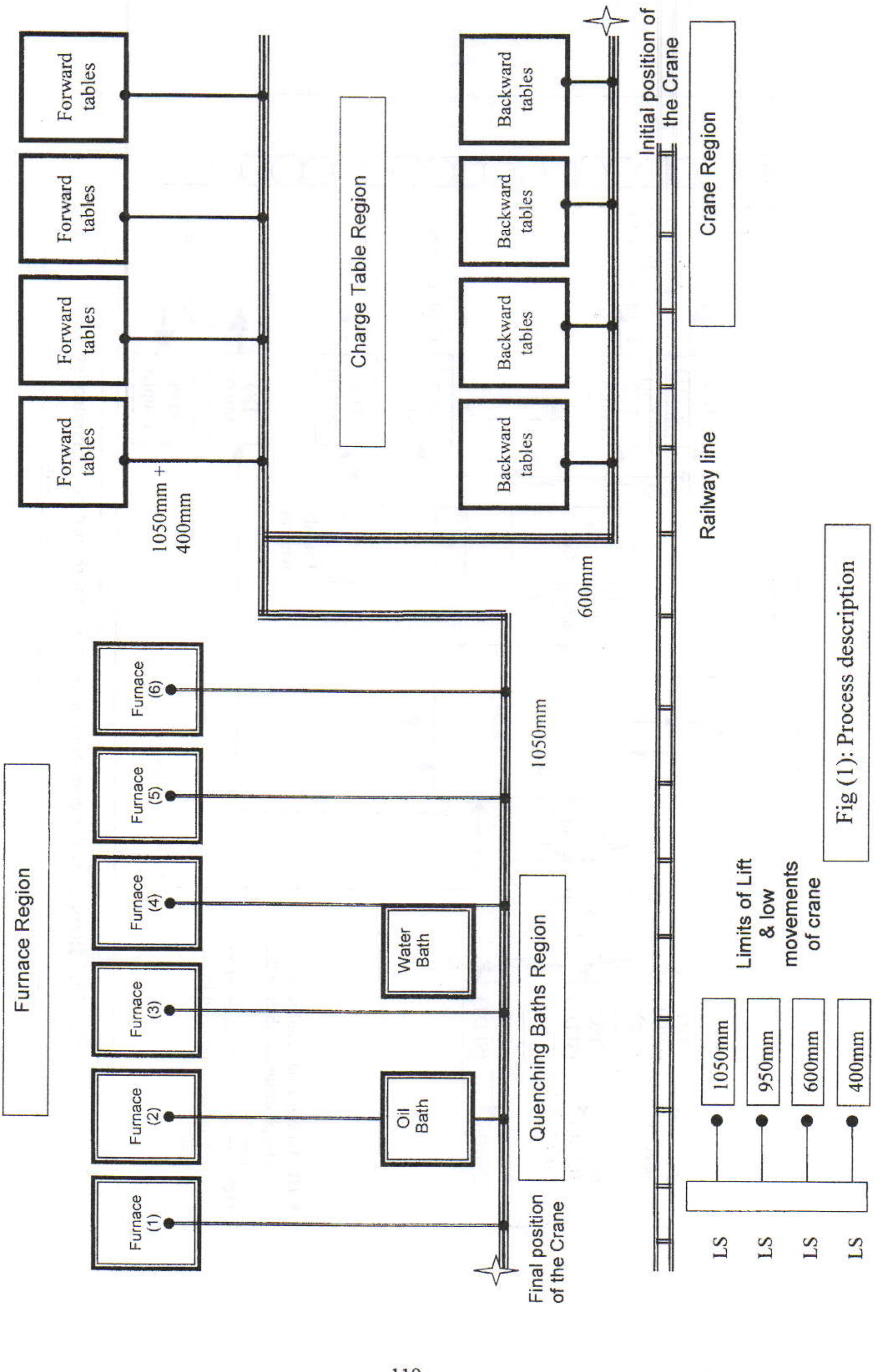
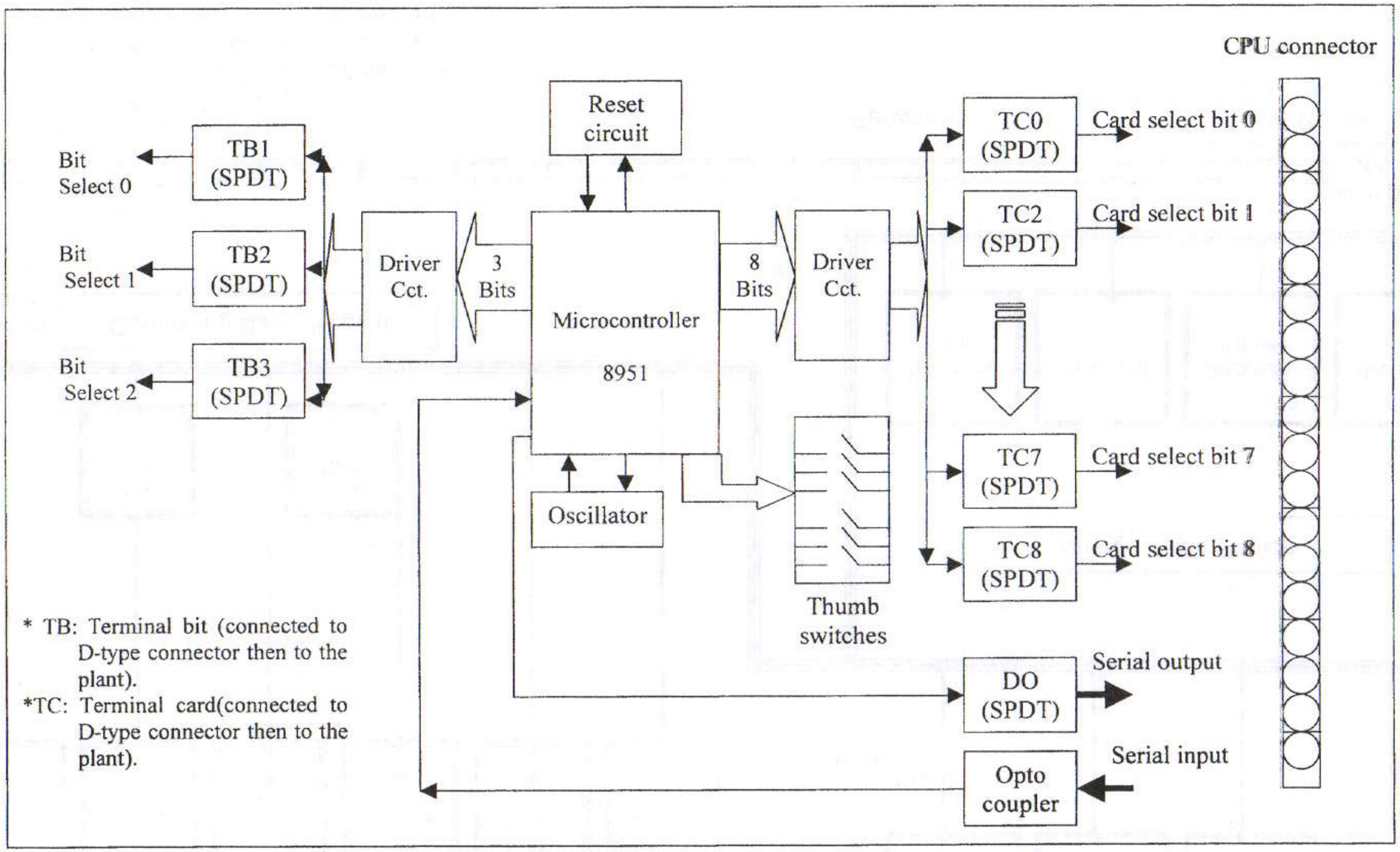


Fig (1): Process description



* TB: Terminal bit (connected to D-type connector then to the plant).
*TC: Terminal card(connected to D-type connector then to the plant).

Fig (2): Microcontroller card connections (motherboard of the control system)

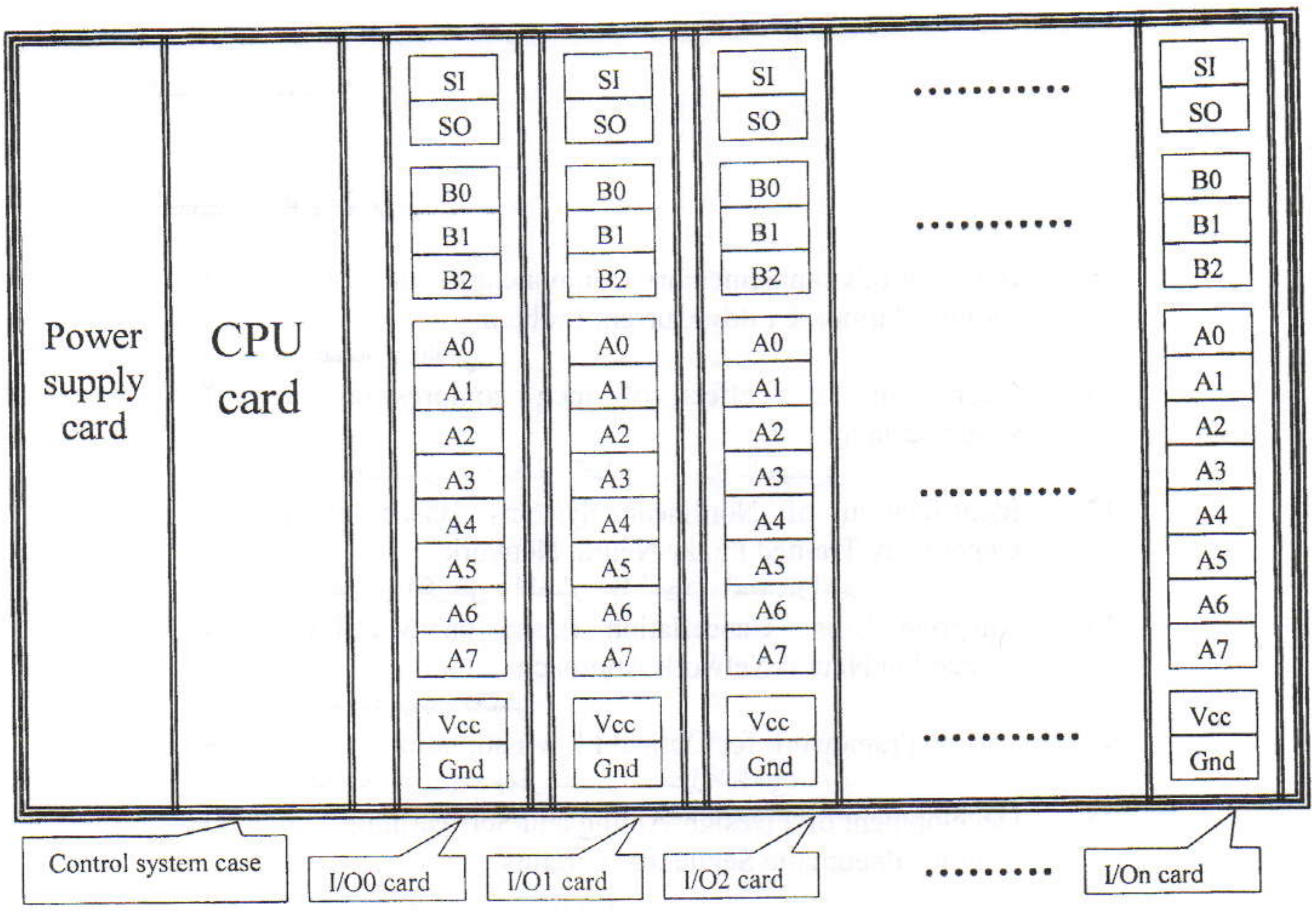
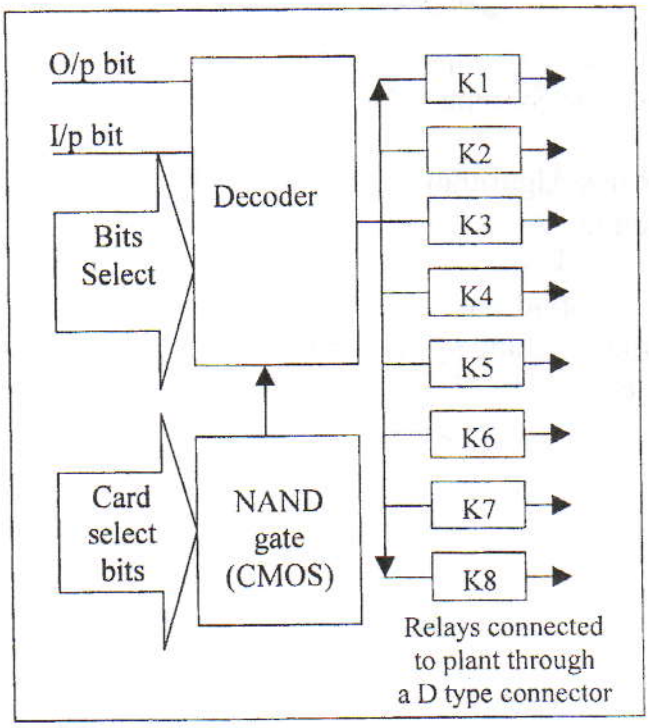
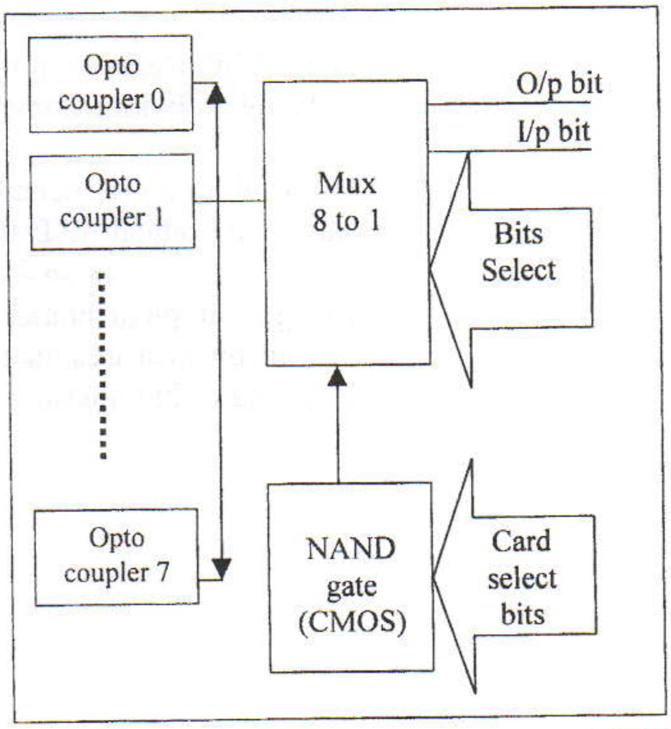


Fig (3): the layout of control system in the cover



Fig(4):the internal structure of the output cards



Fig(5):the internal structure of the Input cards