Title:

Programmer's Reference Manual for
TCP 701, Diagnostic Panel to RC 3603 CPU.

Revision 0.

A/S REGNECENTRALEN

RC SYSTEM LIBRARY: FALKONERALLE 1 DK-2000 COPENHAGEN F

RCSL No:	52-AA542
Edition:	September, 1976
Author:	Knud E. Hansen

Keywords:

TCP 701, RC 3603, Revision 0.

Abstract:

This paper describes the logical structure of the TCP 701, Diagnostic Panel, to the RC 3603 Central Processor Unit.

(20 pages).

Printed by A/S Regnecentralen, Copenhagen

Users of this manual are cautioned that the specifications contained herein are subject to change by RC at any time without prior notice. RC is not responsible for typographical or arithmetic errors which may appear in this manual and shall not be responsible for any damages caused by reliance on any of the materials presented.

CONTENTS Section

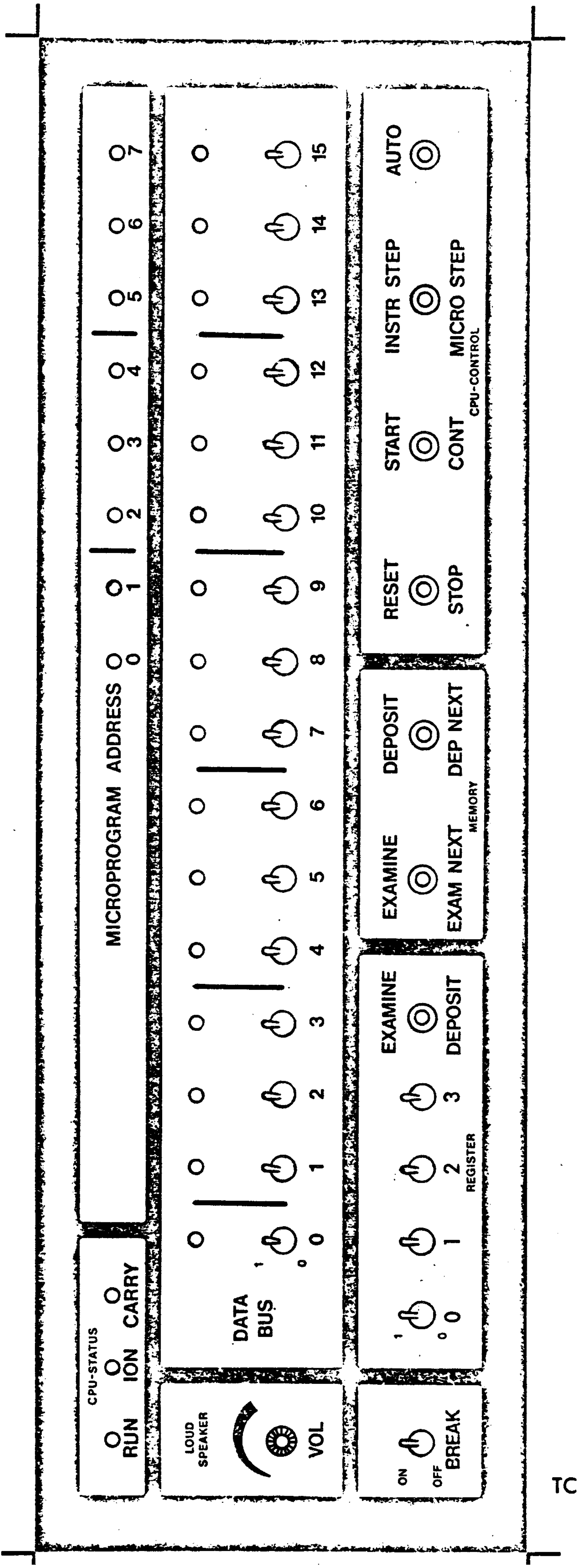
---

pp. 1 : 20.

INTRODUCTION

The Diagnostic Panel to the RC 3603 CPU contains all the function switches and displays all the information needed to operate the CPU. The function and data switches on the console allow the operator to perform many useful operations and the lights reflect the current state of the machine. If a light is lit, it means the corresponding bit is 1. If the light is not lit, the corresponding bit is 0. The lights and their meanings, the switches and their meanings are described in the following sections.

On fig. 1.0 is shown the physical outline of the Diagnostic Panel TCP 701.



DWG NO	DRAWN BY	SCALE
A30132	200976 JD	3 1

Fig. 1

DIAGNOSTIC PANEL LIGHTS 2

The lights reflect the current state of the machine.

RUN. 2.1

The CPU is executing instructions.

ION. 2.2

This light is lit, when the Interrupt On is 1.

CARRY. 2.3

This light is lit, when the Carry bit is 1.

MICRO ADDRESS 0-7. 2.4

These 8 lights display the current state of the microprogram address register. MICRO ADDRESS 7 is the least significant bit of the address.

DATA BUS 0-15. 2.5

These 16 lights display what is currently on the memory bus.

DIAGNOSTIC PANEL SWITCHES 3

Data Switches. 3.1

Beneath the data lights is a row of 16 switches. These switches are used to enter either data or addresses and can be read using the Read Switches instruction. When these switches are in the upper position, they represent a 1; when down, they represent a 0.

Register Select Switches. 3.2

These switches are used during the console functions, REG Deposit and REG Examine, to select one of 16 Registers. When these switches are in the upper position, they represent a 1; when down, they represent a 0.

The figure below indicate the relationship between the switch states and the selected register.

SWITCH				REGISTER
0	1	2	3	
0	0	0	0	AC0
0	0	0	1	AC1
0	0	1	0	AC2
0	0	1	1	AC3
0	1	0	0	PC (Program Counter)
0	1	0	1	BREAK ADDRESS
0	1	1	0	BREAK INSTRUCTION
0	1	1	1	SPARE REGISTERS
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	USED TEMPORARILY DURING SWAP

Fig. 3.2

FUNCTION SWITCHES. 3.3

These switches are spring loaded. When pushed up, they perform the function labelled above the switch, and when pushed down, they perform the function labelled below the switch. When released, these switches return to a neutral "off" position. The switches and their functions are explained in the following sections.

REG EXAM - REG DEP. 3.3.1

When this switch is pushed up, the contents of the register indicated by the current setting of the Register SELECT SWITCHES (section 3.2) are displayed in the Data Bus lights.

When this switch is pushed down, the contents of the Data Switches (section 3.1) are deposited into the register indicated by the current setting of the Register SELECT SWITCHES.

Examine Register 4 (PC) gives the current state of the Program Counter.

EXAMINE - EXAMINE NEXT. 3.3.2

When this switch is pushed up, the EXAMINE function is performed. The address indicated by data switches 1-15 is placed in the program counter. The contents of the word addressed by the program counter are then read and displayed in the data lights. If memory extension is selected, data switches 0-15 determine the address value loaded into the program counter, when this function is performed.

When this switch is pushed down, the EXAMINE NEXT function is performed. The current value of the program counter is incremented by one. The contents of the word addressed by the updated value of the program counter are then read and displayed in the data lights.

DEPOSIT - DEPOSIT NEXT. 3.3.3

When this switch is pushed up, the DEPOSIT function is performed. The current setting of the data switches is placed into the word addressed by the current value of the program counter. The updated value of the altered word is displayed on the data lights.

When this switch is pushed down, the DEPOSIT NEXT function is performed. The program counter is incremented by one and the current setting of the data switches is placed into the word addressed by the updated value of the program counter. The updated value of the altered word is displayed in the data lights.

RESET - STOP. 3.3.4

When this switch is pushed up, the RESET function is performed and an I/O RESET instruction is executed.

The CPU is stopped after completing the current processor cycle. The Interrupt On flag, the 16-bit priority mask, and all Busy and Done flags are set to 0.

When this switch is pushed down, the STOP function is performed. The CPU is stopped after completing the current instruction and before executing the next instruction. If an I/O Device requests an interrupt during the execution of the current instruction, it is honored before the CPU is stopped. Data Channel Requests are still honored, when the CPU is stopped.

If the current instruction contains an infinitely long indirect addressing chain, pressing stop will not stop the CPU. In this situation only pressing RESET will STOP the CPU.

START - CONTINUE. 3.3.5

When this switch is pushed up, the START function is performed.

The address indicated by data switches 1-15 is placed in the program counter and sequential operation of the processor begins with the word addressed by the updated value of the program counter.

When this switch is pushed down, the CONTINUE function is performed. Sequential operation of the processor continues from the current state of the machine.

INSTRUCTION STEP - MICRO STEP. 3.3.6

When this switch is pushed up, the INSTRUCTION STEP function is performed. The instruction contained in the word addressed by the current value of the program counter is executed and then the CPU is stopped. The data displayed by the data lights depends on the instruction as follows:

JMP	Direct → Instruction
LDA, STA	operand
ISZ, DSZ	operand
JSR	Direct → Instruction
	Indirect → Effective Address
Arithmetic and logical	ALU result
IN/OUT	Data

When this switch is pushed down, the MICRO STEP function is performed. The CPU performs a single micro cycle and then stops. The MICRO ADDRESS lights indicate the current micro address.

AUTOLOAD. 3.3.7

When this switch is pushed up, the Program Load function is performed. The contents of the bootstrap read-only memory are placed in memory location 0-37_8 and a "JMP 0" instruction is performed. This switch has no function when pushed down.

BREAK. 3.4

When this switch is in the upper state, the contents of the BREAK ADDRESS register (fig. 3.2) is compared with the contents of the Program Counter whenever an instruction is executed.

If the contents of the BREAK ADDRESS register and the contents of the Program Counter are equal, the normal sequential program execution is stopped, and the next instruction is taken from the BREAK INSTRUCTION register (fig. 3.2). If this instruction is a HALT, the CPU is stopped.

In Break the normal execution time for each instruction is extended with 750 nS.

This feature can be used during program testing to place BREAK POINTS in the program without reassembling the program. In this example the BREAK ADDRESS register should contain the wanted BREAK POINT address and the BREAK INSTRUCTION register a HALT instruction. When BREAK occurs, the contents of PC is:

PC = BREAK ADDRESS + 1.

NOTE: The instruction in the location to which the BREAK ADDRESS points, is not executed if BREAK occurs.

The Break Instruction (AC6) replaces the instruction placed in the location defined by AC5.

LOUDSPEAKER. 3.5

The loudspeaker placed on the Diagnostic Panel is modulated by the value of the carry bit.

The volumen control is used to preset the soundlevel from the loudspeaker.

POWER/LOCK Switch. 3.6

Normally, the TCP 701 panel is used for hardware diagnostic purposes and program testing; when terminating these purposes the panel is removed from the CPU. The TCP 701 is placed in the opened front frame of CHS 702 (in which CPU 708 is installed).

In other configurations TCP 701 is mounted in a frame for rack mounting. This frame contains a POWER/LOCK switch. This switch is a 3-positions key-operated rotary switch that controls power and locks the Diagnostic Panel.

Turning it to ON simply turns on power. Turning it to LOCK keeps power on and disables the operating switches on TCP 701 so no one can interfere with the operation of the processor (the operator can still use the data switches to supply information to the program). The key can be removed in the LOCK state.

PROGRAM LOADING 4

Before a program can be executed, it must be brought into memory. This requires that a loading program already reside in memory. In the event that there is no loading program in memory, a small specialized loading program is normally placed in memory and used to read in the loading program. This small loading program is called a "bootstrap loader". The function of the bootstrap loader is to read in a more general-purpose loading program which can be used to load the user's programs. Two methods are available for entering a bootstrap loader into memory. The operator can either enter it via the data switches and the deposit switch, or, if the computer is so equipped, he can use the program load.

Manual Loading. 4.1

When using an RC 3603 computer without the program load, a bootstrap loader must be entered into memory manually using the switches on the console. On fig. 4.1 is shown a bootstrap loader designed for use with binary loader described in RCSL: 52-AA563 (Paper tape: RCSL: 52-AA564).

This loader reads in a specially formatted tape from the paper tape reader. This type has only 4 bits per frame and the loader assembles these frames into complete words. This bootstrap should be placed in memory starting at that location which is 20_8 less than the highest available memory location. After the bootstrap is entered, start it at location B 770.

Automatic Loading. 4.2

To enter a loader program, the operator must first set up the device that is to be used and set its octal device code into data switches 10-15.

BOOTSTRAP PROGRAM

```
ADDRESS   DATA

B757      126440    READ:   SUBO    1,1
B760      0636DD            SKPDN   DEVICE
B761      000777            JMP     .-1
B762      0605DD            DIAS    0,DEVICE
B763      127100            ADDL    1,1
B764      127100            ADDL    1,1
B765      107003            ADD     0,1,SNC
B766      000772            JMP     READ+1
B767      001400            JMP     0,3
B770      0601DD    BOOT:   NIOS    DEVICE
B771      004766            JSR     READ
B772      044402            STA     1,.+2
B773      004764            JSR     READ
```

B = Two most significant octal digits of the largest available memory address.

(B = 37 for 16K memory,
B = 77 for 32K memory).

DD = Device Code $(10)_8$ when TTY
Device Code $(12)_8$ when PTR

To load the binary loader place the tape in the reader, set the switches to B770 and activate START.

Fig. 4.1

If the device is a data channel device, set data switch 0 to 1. If the device is a low-speed device, set data switch 0 to 0. After this is done, push the AUTO switch to the up position. The bootstrap loader will be deposited into memory locations 0-37_8 and started at location 0. The bootstrap loader is stored in two Read only Memories.

The bootstrap loader reads the data switches, sets up its own I/O instructions with the specified device code, and then performs a program load procedure depending upon the state of data switch 0.

If the switch is a 1, the bootstrap loader starts the device for data channel storage beginning at location 0 and then loops at location 377_8 until a data channel transfer places a word into that location.

After a word has been placed in location 377_8, it is executed as an instruction. Typically, this word is a JUMP into the data that the data channel has placed in the first 377_8 memory locations.

If data switch 0 is a 0, the bootstrap loader reads the loader program via programmed I/O. The device must supply 8-bit data bytes, and each pair of bytes is stored as a single word in memory; wherein the first and second bytes read become the left and right halves of the word. To simplify the positioning of the tape in the reader, the bootstrap loader ignores leading null characters. It does not begin storing any words until it reads a non-zero synchronization byte. The first word following this synchronization byte must be the negative of the total number of words to be read, including the first word. The number of words to be read, including the first word may not be greater than 192_{10}. The bootstrap loader stores these words beginning at memory location 100_8. After storing the last word read, it transfers control to that location.

Listed in fig. 4.2 is an example of a 32 word bootstrap loader (F02). This program is capable of loading in either of the manners described above. In fig. 4.3 is shown a list of available bootstrap loaders.

The binary loader used by the bootstrap loader in F02 is RCSL: 44-RT551 (paper tape 44-RT550).

```
0001  .MAIN

       000000               .LOC     0

00000 060477 BEG:           READS    0        ; READ SWITCHES INTO ACO
00001 105120                MOVZL    0,1      ; ISOLATE DEVICE CODE
00002 124240                COMOR    1,1      ; -DEVICE CODE -1

00003 010011 LOOP:          ISZ      OP1      ; COUNT DEVICE CODE INTO ALL
00004 010031                ISZ      OP2      ; IO INSTRUCTIONS
00005 010033                ISZ      OP3      ;
00006 010014                ISZ      OP4      ;
00007 125404                INC      1,1,SZR  ; DONE ?
00010 000003                JMP      LOOP     ; NO INCREMENT AGAIN

00011 060077 OP1:           060077            ; START DEVICE;(NIOS 0) -1
00012 030017                LDA      2,C377   ; YES,PUTJMP 377INTO LOCATION 377
00013 051377                STA      2,377    ;
00014 063377 OP4:           063377            ; BUSY ? : ( SKPBN 0 ) -1
00015 000011                JMP      OP1      ; NO, GO TO OP1
00016 101102                MOVL     0,0,SZC  ; LOW SPEED DEVICE?(TEST SWITCH0)
00017 000377 C377:          JMP      377      ; NO,GO TO 377 AND WAIT FOR CHAN.

00020 004031 LOOP2:         JSR      GET+1    ; GET A FRAME
00021 101065                MOVC     0,0,SNR  ; IS IT NONZERO ?
00022 000020                JMP      LOOP2    ; NO, IGNORE AND GET ANOTHER

00023 004030 LOOP4:         JSR      GET      ; YES, GET A FULL WORD
00024 046027                STA      1,@C77   ; STORE STARTING AT 100
00025 010100                ISZ      100      ; COUNT WORD - DONE ?
00026 000023                JMP      LOOP4    ; NO, GET ANOTHER
00027 000077 C77:           JMP      77       ; YES - LOCATION COUNTER AND

                                              ; JUMP TO LAST WORD
00030 126420 GET:           SUBZ     1,1      ; CLEAR AC1, SET CARRY
             OP2:
00031 063577 LOOP3:         063577            ; DONE ? : ( SKPDN 0 ) -1
00032 000031                JMP      LOOP3    ; NO, WAIT
00033 060477 OP3:           060477            ; YES,READ INTO AC0:(DIAS 0,0)-1
00034 107363                ADDCS    0,1,SNC  ; ADD 2 FRAMES SWAPPED-GOTSECOND?
00035 000031                JMP      LOOP3    ; NO, GO BACK AFTER IT.
00036 125300                MOVS     1,1      ; YES, SWAP AC1
00037 001400                JMP      0,3      ; RETURN WITH FULL WORD

                          .END
```

BOOTSTRAP LOADER (F02) FOR
AUTOMATIC PROGRAM LOADING

Fig. 4.2

	Device Type	Bit 0	Device Number Bit 10-15 (octal)
F01	Mag. Tape	1	30
F02	PTR	0	12
F03	CDR	0	16
F04	FDD	0	61
F05	DKP	0	73
F06	ASL		*) NOTE
F08	DSC	0	20
(F01)	FPA	1	46

NOTE: Works together with another program load, i.e. F04.

LIST OF AVAILABLE PROGRAM LOADS

Fig. 4.3

RESET MEMORY

5

If a new memory module is installed there is perhaps parity errors in part of this memory. Before running programs in this memory, correct parity has to be written in the whole memory. To do this use the following program entered into memory manually using the switches of the Diagnostic Panel.

RESET PROGRAM:

```
        ADDRESS    DATA

           0       126520      BEG:     SUBZL     1,1      ;
           1       135100               MOVL      1,3      ;
           2       175500               INCL      3,3      ;
           3       045400      LOOP:    STA       1,03     ;
           4       175400               INC       3,3      ;
           5       0 00003              JMP       LOOP     ;
```

START the program at LOCATION 0. The Reset Program will loop in location 1 when finished.

First at the time this program has been running the CPU is switched to the STOP ON PARITY ERROR mode.

On fig. 6 is shown the connection between CPU 708 and the Diagnostic Panel TCP 701. The TCP 701 is connected to the edge connector 1001 on the CPU board and power supplied from the CPU.

Before connecting TCP 701 to or removing TCP 701 from the CPU, secure that the ENABLE TCP switch on the CPU Front Panel is in the upper state, else the operation may disturbe the program execution.

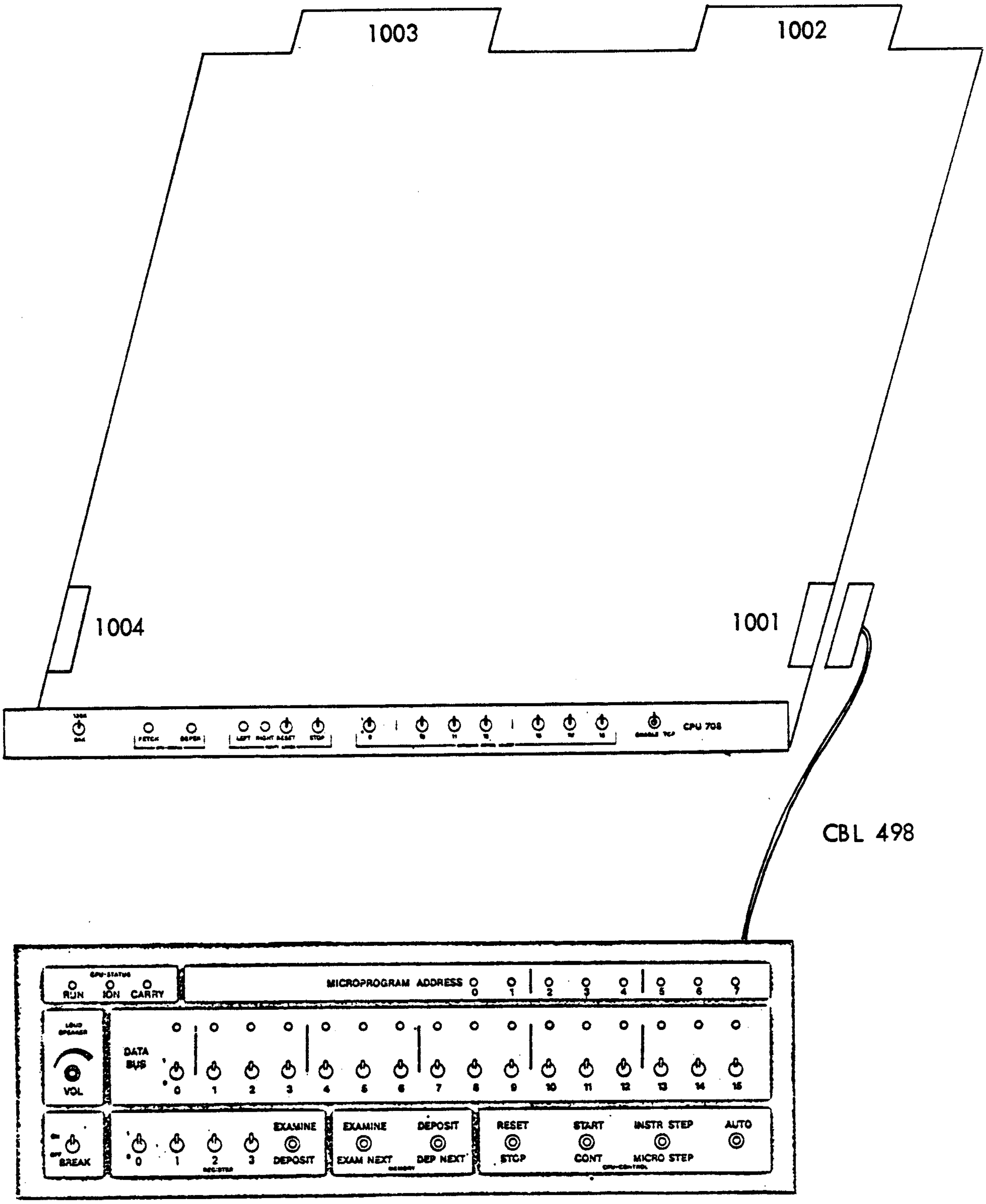


Fig. 6.0: System Outline, Figure.