Microprogramming Accessories
For 2100 Series Minicomputers

12908B Writeable Control Store

12909A PROM Writer
WRITEABLE CONTROL STORE (WCS)

Features
- Incorporates microprograms into computer hardware.
- 256 24-bit word storage for microprograms.
- Enables access to six additional high-speed registers.
- 196 nanosecond cycle time.
- Dynamically alterable.
- Read/Write capability from memory and disc.

MICROPROGRAMMING
A microprogram is a computer program written with advanced, concise instructions that are stored in computer hardware rather than in memory. This makes it possible to execute microprograms from 5 to 10 times faster than ordinary software subroutines. In addition your subroutines can be coded using a fraction of the instruction words needed for conventional programming. Microprograms are expressed in 24-bit, 6 field format. These powerful instructions are executed in a mere 196 nanoseconds. All this not only saves vast memory space, but reduces execution time to a fraction of conventional computing time. Microprogramming enables you to use six additional high-speed registers for fast execution of arithmetic and logic functions and it provides your routines with the ultimate in security by storing them in the hardware.

Thus your own specific programming tasks are performed in the most efficient state-of-the-art manner in a fraction of the time formerly required. Computer applications that are unique to your needs can be added by microprograms to customize the 2100 to your own specific requirements. As your computer needs develop, the 2100 can be modified with different microprograms, thus extending the life and capability of your computer system.

WRITEABLE CONTROL STORE
The Writeable Control Store (WCS) Card contains semi-conductor random access memory for storage of microprograms. Each card contains 256 24-bit words. Up to three WCS cards can be inserted in the computer mainframe.

The instruction word format as stored on the WCS is:

<table>
<thead>
<tr>
<th>Bits</th>
<th>23-21</th>
<th>20-17</th>
<th>16-12</th>
<th>11-8</th>
<th>7-4</th>
<th>3-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field</td>
<td>R-Bus</td>
<td>S-Bus</td>
<td>Function</td>
<td>Store</td>
<td>Special</td>
<td>Skip</td>
</tr>
</tbody>
</table>

You can even have your own library of microprograms on disc and transfer them to the WCS card as needed. After execution, the microprograms can be swapped with other microprograms on the disc or left on the WCS card.

By such use of microprograms and disc files, you can dynamically alter the 2100, configuring its power and speed to your specific purposes. The computer can thus be customized to entirely different system functions as quickly as microprograms can be transferred from disc files.

The microprograms that will not change can be permanently fused on read-only memory chips by the Programmable Read-Only Memory (PROM) Writer.

PROGRAMMABLE READ-ONLY MEMORY (PROM) WRITER
The Programmable Read-Only Memory Writer consists of one card that is temporarily inserted into a 2100 I/O slot and a small box in which the read-only memory chips are fused. Your microprograms can be transferred from memory to PROM chips by the PROM Writer.

The software provided with the PROM Writer enables the system operator to control the fusing. Microprogrammed chips are automatically verified and any missed bits are immediately fused.
The figure given below illustrates the sequence of microprogram implementation.

SOFTWARE
A complete microprogramming software package is supplied with the WCS cards and PROM Writer. It includes the microprogram assembler, drivers, and Debug Editors for both the HP Basic Control System and HP Disc Operating System. Microprograms can be called from Assembly Language, FORTRAN I and IV, ALGOL, and HP Extended BASIC.

A detailed explanation of microprogramming methods and instructions is provided in HP 2100 Computer Microprogramming Guide (5951-3028) and HP 2100 Computer Microprogramming Software (02100-90133).

The calling sequences from Assembly Language and FORTRAN are given below as well as a sample microprogram.

Passing Control From An Assembly Language Program
There are two ways to pass control from an assembly language program to a microprogram. The first applies only if the RAM (Random Access Memory) pseudo-instruction is available; the second applies in any case. The two methods are illustrated below.

Passing Control From an Assembly Language Program to a Microprogram
Method 1: RAM SWB

Method 2: SWB EQU xxxB

Passing Control From A Fortran Program
A FORTRAN program passes control to a microprogram indirectly by way of an assembly language program.

For example, the FORTRAN statement

CALL XYZ (A,B)

generates the following calling sequence:

JSB XYZ
DEF *+3
DEF address of the first parameter
DEF address of the second parameter

When the above calling sequence is executed, control passes to the assembly language program named XYZ. XYZ replaces the JSB XYZ instruction in the above calling sequence with the 105xxx macro instruction and then passes control to the 105xxx macro instruction. The program XYZ is as follows:

ENT XYZ
XYZ NOP
LDA XYZ
ADD =D—1
LDB 105xxx
STB 0,1
JMP 0,1

Block Move Microprogram
This sample microprogram moves a group of computer words from one place in core memory to another. When the microprogram receives control, it is assumed that:

- The number of words to be moved is in the A-register (in two’s complement form).
- The FROM address is in the B-register.
- The TO address is in the core memory location pointed to by the P-register.

The HP assembly language calling sequence is as follows:

LDA (number-of-words)
LDB from-address
105xxx
DEF to-address (cannot be indirect)

The RAM pseudo-instruction automatically forms the 105xxx macro instruction using the constant supplied in the EQU statement (the 105xxx macro instruction replaces the RAM pseudo-instruction). In both cases, xxx is 000-377.

Block Move Microprogram
Microinstruction Commentary

**Microprogram Mnemonics**

<table>
<thead>
<tr>
<th>R-bus</th>
<th>S-bus</th>
<th>Function</th>
<th>Store</th>
<th>Special</th>
<th>Skip</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOP</td>
<td>NOP</td>
<td>IOR</td>
<td>NOP</td>
<td>NOP</td>
<td>NOP</td>
</tr>
<tr>
<td>A</td>
<td>P</td>
<td>XOR</td>
<td>M</td>
<td>CNTR</td>
<td>EOP</td>
</tr>
<tr>
<td>B</td>
<td>S1</td>
<td>AND</td>
<td>T</td>
<td>CW</td>
<td>COUT</td>
</tr>
<tr>
<td>Q</td>
<td>S2</td>
<td>NOR</td>
<td>I1</td>
<td>ECYN</td>
<td>CTR</td>
</tr>
<tr>
<td>F</td>
<td>S3</td>
<td>LWF</td>
<td>A</td>
<td>ECY2</td>
<td>CTR1</td>
</tr>
<tr>
<td>AAB</td>
<td>S4</td>
<td>ARS</td>
<td>B</td>
<td>IOG1</td>
<td>FLG</td>
</tr>
<tr>
<td>CAB</td>
<td>M</td>
<td>LGS</td>
<td>Q</td>
<td>L1</td>
<td>ICTR</td>
</tr>
<tr>
<td>CQ</td>
<td>T</td>
<td>CRS</td>
<td>F</td>
<td>R1</td>
<td>NEG</td>
</tr>
<tr>
<td>ADR</td>
<td>JMP</td>
<td>P</td>
<td>RSS</td>
<td>NMPV</td>
<td></td>
</tr>
<tr>
<td>CNTR</td>
<td>CJMP</td>
<td>S1</td>
<td>RW</td>
<td>ODD</td>
<td></td>
</tr>
<tr>
<td>CL</td>
<td>JSB</td>
<td>S2</td>
<td>AAB</td>
<td>OVF</td>
<td></td>
</tr>
<tr>
<td>CR</td>
<td>RSB</td>
<td>S3</td>
<td>ASG1</td>
<td>RPT</td>
<td></td>
</tr>
<tr>
<td>CIR</td>
<td>ADD</td>
<td>S4</td>
<td>ASG2</td>
<td>TBZ</td>
<td></td>
</tr>
<tr>
<td>IOI</td>
<td>ADDO</td>
<td>IOO</td>
<td>LEP</td>
<td>UNC</td>
<td></td>
</tr>
<tr>
<td>RRS</td>
<td>SUB</td>
<td>AAB</td>
<td>SRG1</td>
<td>AAB</td>
<td></td>
</tr>
<tr>
<td>COND</td>
<td>MPY</td>
<td>CAB</td>
<td>SRG2</td>
<td>NAAB</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DIV</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DEC</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>INC</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>INCO</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SOV</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CLO</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SFLG</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CFLG</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RFE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RFI</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PIA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**SPECIFICATIONS**

**WRITEABLE CONTROL STORE CARD**

**PHYSICAL**

Dimensions
- Width: 7-3/4 inches (196 millimeters)
- Height: 8-11/16 inches (222 millimeters)

**CURRENT REQUIRED**

+4.85 volts supply 4.6 amperes
+2.00 volts supply 0.15 amperes

**EQUIPMENT SUPPLIED**

- 12908B Writeable Control Store PCA (12908-60006)
- Jumper Board Assembly (12908-60003)
- Operating and Service Manuals (12908-90001)

**PROGRAMMABLE READ ONLY MEMORY WRITER**

**PHYSICAL**

- Width: 7-3/4 inches (196 millimeters)
- Height: 8-11/16 inches (222 millimeters)

**Prom Writer Mounting Fixture**

- Width: 3 inches (76. millimeters)
- Height: 5-3/4 inches (146 millimeters)
- Depth: 1-1/2 inches (38 millimeters)

**CURRENT REQUIRED**

+5 volt supply ≤ 0.500 amperes
-2 volt supply ≤ 0.040 amperes
+12 volt supply ≤ 0.250 amperes
+30 volt supply ≤ 0.250 amperes

**EQUIPMENT SUPPLIED**

- 12909A PROM Writer Printed-circuit Assembly (12909-60001)
- PROM Writer Mounting Fixture (12909-60002)
- Test ROM (1816-0250)
- Operating and Service Manual (12909-90001)