HEWLETT
PACKARD

HEWLETT-PACKARD COMPANY
LOGIC SYSTEMS DIVISION

HP 64000 Logic Development System

SYSTEM RELEASE BULLETIN

Part Number: 5958-6019
E0886

Printed: AUGUST 1986

```
 SSSSS     RRRRRR     BBBBBB
S     S    R     R    B     B
S          R     R    B     B
 SSSSS     RRRRRR     BBBBBB
      S    R   R      B     B
S     S    R    R     B     B
 SSSSS     R     R    BBBBBB
```

SYSTEM RELEASE BULLETIN

64000 Logic Development System

AUGUST 1986

This System Release Bulletin (SRB) documents all fixes and enhancements that are incorporated in the latest release of software for the 64000 Logic Development System.

The SRB is provided as a benefit of Hewlett-Packard's Software Support Services.

The five sections of the SRB are:

SOFTWARE RELEASE CONTENTS - lists the new revision codes for the 64000 products.

PRODUCT INDEX - lists product names and numbers which are included in this issue.

KPR NUMBER INDEX - sequential list of SR numbers.

KEYWORD INDEX - brief description of each SR.

KNOWN PROBLEM REPORTS - the actual reports.

Software release contents

Product name		Product number	uu.ff
*6805U/R&P EMULATION		64192	01.07
*6805U/R&P EMULATION		64193	01.07
*6805U/R&P EMULATION		64194	01.07
*8051 ASSEMB		64855	01.07
*8051 ASSEMB	300	64855S004	01.10
*8051 ASSEMB	500	64855S001	01.40
*8051 ASSEMB	VAX	64855S003	01.50
*8086/8 ASSEMB		64853	02.01
*8086/8 ASSEMB	300	64853S004	02.10
*8086/8 ASSEMB	500	64853S001	02.20
*8086/8 ASSEMB	VAX	64853S003	02.30
*8096 ASSEMB		64860	01.02
*8096 ASSEMB	300	64860S004	01.10
*8096 ASSEMB	500	64860S001	01.20
*8096 ASSEMB	VAX	64860S003	01.30
*PROM PROGRAMMER		64500	01.09
*USER DEF ASSEMB	500	64851S001	01.40
*USER DEF ASSEMB	VAX	64851S003	01.50

Product index

Product name		Product number	Page
6805U/R&P EMULATION		64192	1
8051 ASSEMB		64855	2
8051 ASSEMB	300	64855S004	3
8051 ASSEMB	500	64855S001	4
8051 ASSEMB	VAX	64855S003	6
8086/8 ASSEMB		64853	8
8086/8 ASSEMB	300	64853S004	9
8086/8 ASSEMB	500	64853S001	10
8086/8 ASSEMB	VAX	64853S003	12
8096 ASSEMB	500	64860S001	14
8096 ASSEMB	VAX	64860S003	15
USER DEF ASSEMB	500	64851S001	16
USER DEF ASSEMB	VAX	64851S003	17

Report number index

Report #	page	Report #	page	Report #	page	Report #	page
1650006536	17	D200048405	16	D200049387	16	D200049577	15
5000129304	2	D200048413	17	D200049395	19	D200050906	9
D200019869	16	D200048439	12	D200049411	11	D200052324	4
D200019877	17	D200048488	4	D200049429	13	D200052332	6
D200028555	8	D200048496	6	D200049478	5	D200052340	3
D200028589	10	D200048579	14	D200049486	7	D200053140	1
D200028597	12	D200048587	15	D200049569	14	D200053504	18
D200048157	10						

Keyword index

- 6805U/R&P EMULATION -

Keyword	Product number	uu.ff	Description	Report #	page
********none********	64192	01.06	EBPP will not disassemble 6805 code.	D200053140	1

- 8051 ASSEMB -

Keyword	Product number	uu.ff	Description	Report #	page
CODE GENERATOR	64855	01.05	Incorrect opcode "MOV A,ACC" allowed by our assembler	5000129304	2

- 8051 ASSEMB -

Keyword	Product number	uu.ff	Description	Report #	page
CODE GENERATOR	64855S004	01.00	Incorrect opcode "MOV A,ACC" allowed by our assembler	D200052340	3

- 8051 ASSEMB -

Keyword	Product number	uu.ff	Description	Report #	page
********none********	64855S001	00.00	Linker output file should use alternate file extension.	D200049478	5
CODE GENERATOR	64855S001	01.30	Incorrect opcode "MOV A,ACC" allowed by our assembler	D200052324	4
MACRO	64855S001	01.30	Conditional instr. .IF with rational oper. in Macro creates bad code	D200048488	4

- 8051 ASSEMB -

Keyword	Product number	uu.ff	Description	Report #	page
********none********	64855S003	00.00	Linker output file should use alternate file extension.	D200049486	7
CODE GENERATOR	64855S003	01.40	Incorrect opcode "MOV A,ACC" allowed by our assembler	D200052332	6
MACRO	64855S003	01.40	Conditional instr. .IF with rational oper. in Macro creates bad code	D200048496	6

- 8086/8 ASSEMB -

Keyword	Product number	uu.ff	Description	Report #	page
********none********	64853	02.00	MOV mem,data may cause false Legal Range error.	D200028555	8

- 8086/8 ASSEMB -

Keyword	Product number	uu.ff	Description	Report #	page
********none********	64853S004	02.00	MOV mem,data may cause false Legal Range error.	D200050906	9

- 8086/8 ASSEMB -

Keyword	Product number	uu.ff	Description	Report #	page
********none********	64853S001	00.00	Linker output file should use alternate file extension.	D200049411	11
	64853S001	02.00	MOV mem,data may cause false Legal Range error.	D200028589	10
MACRO	64853S001	02.10	Conditional instr. .IF with rational oper. in Macro creates bad code	D200048157	10

Keyword index

- 8086/8 ASSEMB -

Keyword	Product number	uu.ff	Description	Report #	page
********none********	64853S003	00.00	Linker output file should use alternate file extension.	D200049429	13
	64853S003	02.00	MOV mem,data may cause false Legal Range error.	D200028597	12
MACRO	64853S003	02.20	Conditional instr. .IF with rational oper. in Macro creates bad code	D200048439	12

- 8096 ASSEMB -

Keyword	Product number	uu.ff	Description	Report #	page
********none********	64860S001	00.00	Linker output file should use alternate file extension.	D200049569	14
MACRO	64860S001	01.10	Conditional instr. .IF with rational oper. in Macro creates bad code	D200048579	14

- 8096 ASSEMB -

Keyword	Product number	uu.ff	Description	Report #	page
********none********	64860S003	00.00	Linker output file should use alternate file extension.	D200049577	15
MACRO	64860S003	01.20	Conditional instr. .IF with rational oper. in Macro creates bad code	D200048587	15

- USER DEF ASSEMB -5

Keyword	Product number	uu.ff	Description	Report #	page
********none********	64851S001	00.00	Linker output file should use alternate file extension.	D200049387	16
	64851S001	01.10	Code generated differs from code generated on HP 64000.	D200019869	16
MACRO	64851S001	01.30	Conditional instr. .IF with rational oper. in Macro creates bad code	D200048405	16

- USER DEF ASSEMB -V

Keyword	Product number	uu.ff	Description	Report #	page
********none********	64851S003	00.00	Linker output file should use alternate file extension.	D200049395	19
	64851S003	01.10	Code generated differs from code generated on HP 64000.	D200019877	17
	64851S003	01.40	Macro def. including .IF, within a IF causes assembler to stop code gen.	D200053504	18
MACRO	64851S003	01.20	string comparison does not function using conditional .if instr.	1650006536	17
	64851S003	01.40	Conditional instr. .IF with rational oper. in Macro creates bad code	D200048413	17

Number: D200053140 Product: 6805U/R&P EMULATION 64192 01.06

One-line description:
EBPP will not disassemble 6805 code.

Problem:
When using a 6805 emulator with the Emulation Bus Preprocessor (EBPP), the state analyzer trace list does not disassemble the instructions. The section of the listing which is supposed to display the mneumonics is blank. It was discovered that the wrong inverse assembler was included in the EBPP module (ANLY_304_6805) of the emulation software.

Signed off 06/23/86 in release 201.07

Number: 5000129304 Product: 8051 ASSEMB 64855 01.05

Keywords: CODE GENERATOR

One-line description:
Incorrect opcode "MOV A,ACC" allowed by our assembler

Problem:
The instruction "MOV A,ACC" was assemble and emulated by our products; however, the Intel 8051 goes into the weeds at this instrcution.
At first glance the machine code in the asembler listing appears valid (MOV A,ACC ->0000 E5E0), but the bottom of page 8-35 in Intel's microcontroller handbook states: *MOV A,ACC is not a valid instruction.

Neither our manuals nor AMD's user manual mention this instruction.

Temporary solution:
No known temporary solution.

Number: D200052340 Product: 8051 ASSEMB 300 64855S004 01.00

Keywords: CODE GENERATOR

One-line description:
Incorrect opcode "MOV A,ACC" allowed by our assembler

Problem:
The instruction "MOV A,ACC" was assemble and emulated by our products; however, the Intel 8051 goes into the weeds at this instrcution.
At first glance the machine code in the asembler listing appears valid (MOV A,ACC ->0000 E5E0), but the bottom of page 8-35 in Intel's microcontroller handbook states: *MOV A,ACC is not a valid instruction.

Neither our manuals nor AMD's user manual mention this instruction.

Temporary solution:
No known temporary solution.

Signed off 06/23/86 in release 401.10

Number: D200048488 Product: 8051 ASSEMB 500 64855S001 01.30

Keywords: MACRO

One-line description:
Conditional instr. .IF with rational oper. in Macro creates bad code

Problem:
The use of the conditional instruction, .IF, with rational operator (.EQ.,.NE.,.LT.,.GT.,.LE.,.GE.) in a macro functions incorrectly.
The following program demonstrates this problem:

```
       BUG           MACRO             &VAR
                     .IF &VAR .LE. 0 SUB&&&&
                     NOP
                     NOP
       SUB&&&&       NOP
                     NOP
                     MEND

                     BUG  3
                     BUG -1
                     BUG  0
                     END
```

Passing a 3 appears to create correct code, but 0 causes a ML error. Passing -1 to the MACRO creates code which doesn't call the subroutine. This is incorrect since -1 is less than 0. This same problem occured with all the rational operators on all processors. The problem was consistant on the 64000, VAX, and 9000.

Signed off 06/23/86 in release 101.40

Number: D200052324 Product: 8051 ASSEMB 500 64855S001 01.30

Keywords: CODE GENERATOR

One-line description:
Incorrect opcode "MOV A,ACC" allowed by our assembler

Problem:
The instruction "MOV A,ACC" was assemble and emulated by our products; however, the Intel 8051 goes into the weeds at this instrcution.
At first glance the machine code in the asembler listing appears valid (MOV A,ACC ->0000 E5E0), but the bottom of page 8-35 in Intel's microcontroller handbook states: *MOV A,ACC is not a valid instruction.

Neither our manuals nor AMD's user manual mention this instruction.

Temporary solution:
No known temporary solution.

Signed off 06/23/86 in release 101.40

Number: D200049478 Product: 8051 ASSEMB 500 64855S001 00.00

One-line description:
Linker output file should use alternate file extension.

Signed off 06/23/86 in release 101.40

Number: D200048496 Product: 8051 ASSEMB VAX 64855S003 01.40

Keywords: MACRO

One-line description:
Conditional instr. .IF with rational oper. in Macro creates bad code

Problem:
The use of the conditional instruction, .IF, with rational operator (.EQ.,.NE.,.LT.,.GT.,.LE.,.GE.) in a macro functions incorrectly.
The following program demonstrates this problem:

```
       BUG          MACRO            &VAR
                    .IF &VAR .LE. 0 SUB&&&&
                    NOP
                    NOP
       SUB&&&&      NOP
                    NOP
                    MEND

                    BUG  3
                    BUG -1
                    BUG  0
                    END
```

Passing a 3 appears to create correct code, but 0 causes a ML error.
Passing -1 to the MACRO creates code which doesn't call the subroutine.
This is incorrect since -1 is less than 0. This same problem occured with all the rational operators on all processors. The problem was consistant on the 64000, VAX, and 9000.

Signed off 06/23/86 in release 301.50

Number: D200052332 Product: 8051 ASSEMB VAX 64855S003 01.40

Keywords: CODE GENERATOR

One-line description:
Incorrect opcode "MOV A,ACC" allowed by our assembler

Problem:
The instruction "MOV A,ACC" was assemble and emulated by our products; however, the Intel 8051 goes into the weeds at this instrcution.
At first glance the machine code in the asembler listing appears valid (MOV A,ACC ->0000 E5E0), but the bottom of page 8-35 in Intel's microcontroller handbook states: *MOV A,ACC is not a valid instruction.

Neither our manuals nor AMD's user manual mention this instruction.

Temporary solution:
No known temporary solution.

Signed off 06/23/86 in release 301.50

Number: D200049486 Product: 8051 ASSEMB VAX 64855S003 00.00

One-line description:
Linker output file should use alternate file extension.

Signed off 06/23/86 in release 301.50

Number: D200028555 Product: 8086/8 ASSEMB 64853 02.00

One-line description:
MOV mem,data may cause false Legal Range error.

Problem:
The following code causes a Legal Range error which should not occur:

```
"processor name"
  ASSUME CS:PROG
  ASSUME DS:DATA
  PROG
  MOV MEMORY,#01H
  MOV MEMORY,#80H
             ^Legal Range
  DATA
MEMORY  DBS 1
```

Temporary solution:
Use the corresponding negative number as immediate data:

```
   MOV MEMORY,#-128H
```

Signed off 06/23/86 in release 302.01

Number: D200050906 Product: 8086/8 ASSEMB 300 64853S004 02.00

One-line description:
MOV mem,data may cause false Legal Range error.

Problem:
The following code causes a Legal Range error which should not occur:

```
"processor name"
  ASSUME CS:PROG
  ASSUME DS:DATA
  PROG
  MOV MEMORY,#01H
  MOV MEMORY,#80H
             ^Legal Range
  DATA
MEMORY  DBS 1
```

Temporary solution:
Use the corresponding negative number as immediate data:

```
   MOV MEMORY,#-128H
```

Signed off 06/23/86 in release 402.10

Number: D200028589 Product: 8086/8 ASSEMB 500 64853S001 02.00

One-line description:
MOV mem,data may cause false Legal Range error.

Problem:
The following code causes a Legal Range error which should not occur:

```
"processor name"
  ASSUME CS:PROG
  ASSUME DS:DATA
  PROG
  MOV MEMORY,#01H
  MOV MEMORY,#80H
             ^Legal Range
  DATA
MEMORY  DBS 1
```

Temporary solution:
Use the corresponding negative number as immediate data:

```
   MOV MEMORY,#-128H
```

Signed off 06/23/86 in release 102.20

Number: D200048157 Product: 8086/8 ASSEMB 500 64853S001 02.10

Keywords: MACRO

One-line description:
Conditional instr. .IF with rational oper. in Macro creates bad code

Problem:
The use of the conditional instruction, .IF, with rational operator (.EQ.,.NE.,.LT.,.GT.,.LE.,.GE.) in a macro functions incorrectly.
The following program demonstrates this problem:

```
       BUG             MACRO            &VAR
                       .IF &VAR .LE. 0 SUB&&&&
                       NOP
                       NOP
       SUB&&&&         NOP
                       NOP
                       MEND

                       BUG  3
                       BUG -1
                       BUG  0
                       END
```

Passing a 3 appears to create correct code, but 0 causes a ML error.
Passing -1 to the MACRO creates code which doesn't call the subroutine.
This is incorrect since -1 is less than 0. This same problem occured with all the rational operators on all processors. The problem was consistant on the 64000, VAX, and 9000.

Signed off 06/23/86 in release 102.20

Number: D200049411 Product: 8086/8 ASSEMB 500 64853S001 00.00

One-line description:
Linker output file should use alternate file extension.

Signed off 06/23/86 in release 102.20

Number: D200028597 Product: 8086/8 ASSEMB VAX 64853S003 02.00

One-line description:
MOV mem,data may cause false Legal Range error.

Problem:
The following code causes a Legal Range error which should not occur:

```
"processor name"
  ASSUME CS:PROG
  ASSUME DS:DATA
  PROG
  MOV MEMORY,#01H
  MOV MEMORY,#80H
            ^Legal Range
  DATA
MEMORY  DBS 1
```

Temporary solution:
Use the corresponding negative number as immediate data:

```
   MOV MEMORY,#-128H
```

Signed off 06/23/86 in release 302.30

Number: D200048439 Product: 8086/8 ASSEMB VAX 64853S003 02.20

Keywords: MACRO

One-line description:
Conditional instr. .IF with rational oper. in Macro creates bad code

Problem:
The use of the conditional instruction, .IF, with rational operator (.EQ.,.NE.,.LT.,.GT.,.LE.,.GE.) in a macro functions incorrectly. The following program demonstrates this problem:

```
       BUG           MACRO            &VAR
                     .IF &VAR .LE. 0 SUB&&&&
                     NOP
                     NOP
       SUB&&&&       NOP
                     NOP
                     MEND

                     BUG  3
                     BUG -1
                     BUG  0
                     END
```

Passing a 3 appears to create correct code, but 0 causes a ML error. Passing -1 to the MACRO creates code which doesn't call the subroutine. This is incorrect since -1 is less than 0. This same problem occured with all the rational operators on all processors. The problem was consistant on the 64000, VAX, and 9000.

Signed off 06/23/86 in release 302.30

Number: D200049429 Product: 8086/8 ASSEMB VAX 64853S003 00.00

One-line description:
Linker output file should use alternate file extension.

Signed off 06/23/86 in release 302.30

Number: D200048579 Product: 8096 ASSEMB 500 64860S001 01.10

Keywords: MACRO

One-line description:
Conditional instr. .IF with rational oper. in Macro creates bad code

Problem:
The use of the conditional instruction, .IF, with rational operator (.EQ.,.NE.,.LT.,.GT.,.LE.,.GE.) in a macro functions incorrectly.
The following program demonstrates this problem:

```
BUG           MACRO          &VAR
              .IF &VAR .LE. 0 SUB&&&&
              NOP
              NOP
SUB&&&&       NOP
              NOP
              MEND

              BUG  3
              BUG -1
              BUG  0
              END
```

Passing a 3 appears to create correct code, but 0 causes a ML error.
Passing -1 to the MACRO creates code which doesn't call the subroutine.
This is incorrect since -1 is less than 0. This same problem occured with all the rational operators on all processors. The problem was consistant on the 64000, VAX, and 9000.

Signed off 06/23/86 in release 101.20

Number: D200049569 Product: 8096 ASSEMB 500 64860S001 00.00

One-line description:
Linker output file should use alternate file extension.

Signed off 06/23/86 in release 101.20

Number: D200048587 Product: 8096 ASSEMB VAX 64860S003 01.20

Keywords: MACRO

One-line description:
Conditional instr. .IF with rational oper. in Macro creates bad code

Problem:
The use of the conditional instruction, .IF, with rational operator (.EQ.,.NE.,.LT.,.GT.,.LE.,.GE.) in a macro functions incorrectly. The following program demonstrates this problem:

```
       BUG             MACRO            &VAR
                       .IF &VAR .LE. 0 SUB&&&&
                       NOP
                       NOP
       SUB&&&&         NOP
                       NOP
                       MEND

                       BUG  3
                       BUG -1
                       BUG  0
                       END
```

Passing a 3 appears to create correct code, but 0 causes a ML error. Passing -1 to the MACRO creates code which doesn't call the subroutine. This is incorrect since -1 is less than 0. This same problem occured with all the rational operators on all processors. The problem was consistant on the 64000, VAX, and 9000.

Signed off 06/23/86 in release 301.30

Number: D200049577 Product: 8096 ASSEMB VAX 64860S003 00.00

One-line description:
Linker output file should use alternate file extension.

Signed off 06/23/86 in release 301.30

Number: D200019869 Product: USER DEF ASSEMB 500 64851S001 01.10

One-line description:
Code generated differs from code generated on HP 64000.

Signed off 06/23/86 in release 101.40

Number: D200048405 Product: USER DEF ASSEMB 500 64851S001 01.30

Keywords: MACRO

One-line description:
Conditional instr. .IF with rational oper. in Macro creates bad code

Problem:
The use of the conditional instruction, .IF, with rational operator (.EQ.,.NE.,.LT.,.GT.,.LE.,.GE.) in a macro functions incorrectly. The following program demonstrates this problem:

```
       BUG             MACRO            &VAR
                       .IF &VAR .LE. 0 SUB&&&&
                       NOP
                       NOP
       SUB&&&&         NOP
                       NOP
                       MEND

                       BUG  3
                       BUG -1
                       BUG  0
                       END
```

Passing a 3 appears to create correct code, but 0 causes a ML error. Passing -1 to the MACRO creates code which doesn't call the subroutine. This is incorrect since -1 is less than 0. This same problem occured with all the rational operators on all processors. The problem was consistant on the 64000, VAX, and 9000.

Signed off 06/23/86 in release 101.40

Number: D200049387 Product: USER DEF ASSEMB 500 64851S001 00.00

One-line description:
Linker output file should use alternate file extension.

Signed off 06/23/86 in release 101.40

Number: 1650006536 Product: USER DEF ASSEMB VAX 64851S003 01.20

Keywords: MACRO

One-line description:
string comparison does not function using conditional .if instr.

Problem:

Hosted Macro assembler on Vax does not expand macros properly. The problem is related with "String unequality comparison".

```
              BEGIN         MACRO        &P1
                            .IF &P1 .NE. "" FIN
                            MOV    A,#0FH
              FIN           .NOP
                            MEND

                            BEGIN    MYLABEL
                            BEGIN    ""
                            END
```

The HP64100 allows checking for optional macro parameters by the above example. This method only works with the null ("") operand. If any other string is used for the operand, quotes must be placed either around the parameter at the macro call or around the &P1 in the .IF statement. However, the vax and 9000 do not produce the same code as the HP64100. Although the VAX/9000 does not generate an error message, the code generated is incorrect. For example, the call "BEGIN   MYLABEL" in the above test program creates the following listing.

```
              11            BEGIN    MYLABEL
              +             .IF MYLABEL .NE. "" FIN
              +             MOV A,#0FH
             12             etc.
```

Temporary Solution:

```
             Replace        .IF &P1 .NE. "" FIN
             with           .IF "&P1" .NE. "''" FIN
```

Signed off 06/23/86 in release 301.50

Number: D200019877 Product: USER DEF ASSEMB VAX 64851S003 01.10

One-line description:
Code generated differs from code generated on HP 64000.

Signed off 06/23/86 in release 301.50

Number: D200048413 Product: USER DEF ASSEMB VAX 64851S003 01.40

Keywords: MACRO

One-line description:
Conditional instr. .IF with rational oper. in Macro creates bad code

Problem:
The use of the conditional instruction, .IF, with rational operator (.EQ.,.NE.,.LT.,.GT.,.LE.,.GE.) in a macro functions incorrectly. The following program demonstrates this problem:

```
       BUG            MACRO            &VAR
                      .IF &VAR .LE. 0 SUB&&&&
                      NOP
                      NOP
       SUB&&&&        NOP
                      NOP
                      MEND

                      BUG  3
                      BUG -1
                      BUG  0
                      END
```

Passing a 3 appears to create correct code, but 0 causes a ML error. Passing -1 to the MACRO creates code which doesn't call the subroutine. This is incorrect since -1 is less than 0. This same problem occured with all the rational operators on all processors. The problem was consistant on the 64000, VAX, and 9000.

Signed off 06/23/86 in release 301.50

Number: D200053504 Product: USER DEF ASSEMB VAX 64851S003 01.40

One-line description:
Macro def. including .IF, within a IF causes assembler to stop code gen.

Problem:
If you have a ".IF" in a macro definition and that macro definition is within a conditional assembly "IF" then no code is generated. The program provided demonstrates the problem (see submitter text).

Temporary solution:
Pull the macro definition outside of the conditional if. No code will be generated for the definition.

```
"processor name"

ESSAI       EQU     0

MAC         MACRO
            .IF     ESSAI.EQ.0   FIN
LABEL       LD      A,0
FIN         MEND


            IF      ESSAI
            MAC
            ENDIF

START       LD      A,3
```

Signed off 06/23/86 in release 301.50

Number: D200049395 Product: USER DEF ASSEMB VAX 64851S003 00.00

One-line description:
Linker output file should use alternate file extension.

Signed off 06/23/86 in release 301.50

HEWLETT
PACKARD

5958-6019, August 1986
E0886

Printed in U.S.A.