The DP8344 BCP® Inverse Assembler

OVERVIEW

The DP8344 BCP Inverse Assembler is a software package for use in a Hewlett Packard Logic Analyzer. It was developed by National Semiconductor's Arlington Design Center to allow disassembly of the DP8344 op-code mnemonics.

When developing systems using a RISC processor such as the DP8344, the need often arises to know the sequence of events that occurred in real time in the system. The actual execution flow that occurred in the system can be determined by monitoring the states on the Instruction memory Address bus and the Instruction memory bus of the DP8344 with a Hewlett Packard Logic Analyzer. The DP8344 BCP Inverse Assembler enhances this development tool by displaying the BCP instruction op-code mnemonics on the logic analyzer's screen. This Application Note lists the equipment needed as well as the necessary information to set up, use, and obtain the DP8344 BCP Inverse Assembler. Additionally, the source code flow chart for the DP8344 BCP Inverse Assembler is provided in Appendix A of this Application Note.

EQUIPMENT REQUIRED

The following equipment is required to use the DP8344 BCP Inverse Assembler:

1. DP8344 BCP Inverse Assembler; Available from National Semiconductor.

2. HP1650A or HP1651A Logic Analyzer, or HP16500A Logic Analysis System with an HP16510A State/Timing Card installed.

3. DP8344 Biphase Communications Processor in a System.

It is assumed that the reader is familiar with the operation of the HP Logic Analyzer. For further information refer to the Operation Reference Manual provided with the HP1650A or 1651A Logic Analyzers, or with the HP16510A Logic Analyzer Module. Information pertaining to the operation of the logic analyzer in a state mode will be useful.

SYSTEM SETUP

A block diagram of the setup of the system for using the DP8344 BCP Inverse Assembler is shown in Figure 1. The target system refers to a system containing a BCP which is running. The DP8344 BCP Inverse Assembler is software which has been loaded into the HP Logic Analyzer. The target system is interfaced to the DP8344 BCP Inverse Assembler through the HP Logic Analyzer's channels.

An example of a target system is a Multi-Protocol Adapter (MPA™) installed in a personal computer. The MPA Design/Evaluation Kit includes both the hardware and software that allows the MPA to emulate a 3270 or 5250 display terminal and to support industry standard PC emulation software. The MPA Design/evaluation Kit is available from National Semiconductor (Part No. D88344MPA-EB). All the examples in this document were generated using an MPA board and it's associated software for the target system.

Additional equipment which one may find useful includes an extender card and an 84-pin PLCC Adapter. The extender card brings a PC board out of the PC chasis, allowing easier access to the BCP. An 84-pin PLCC Adapter allows one to directly connect the channels of the logic analyzer to the pins on the BCP. Emulation Technology, Inc., makes an 84-pin PLCC Adapter which it calls a BUG KATCHER. (It is Part No. BC-4-084-PCC5-00000).

The sample target system described above includes the following equipment:

1. IBM® Personal Computer or compatible

2. MPA Development Kit

3. Extender Card (optional)

4. 84-Pin PLCC Adapter

The DP8344 BCP Inverse Assembler requires information from both the Instruction memory Address bus and the Instruction memory data bus of the BCP in the target system. Thus, these pins must be connected to the logic analyzer. The 84-pin PLCC Adapter allows one to directly connect the logic analyzer channels to the BCP. Figure 2 provides a detailed view of the pin connections from the DP8344 to the logic analyzer. The pins can be connected to any of the pods as long as the channel and label definitions are defined accordingly in the FORMAT Menu as described later in this Application Note.

STARTING THE DP8344 BCP INVERSE ASSEMBLER

Once the system hardware has been set up, the DP8344 BCP Inverse Assembler software needs to be installed in the HP Logic Analyzer. The 3½ inch diskette provided in the DP8344 BCP Inverse Assembler Package contains the software for the HP Logic Analyzer. Load the DP8344 BCP Inverse Assembler Software into the HP Logic Analyzer by selecting either LOAD ALL from file BCP, or LOAD State/Timing E, from File BCP.E as in Figure 3. This automatically loads the DP8344 BCP Inverse Assembler as well as the stored State/Timing configuration into the HP Logic Analyzer.
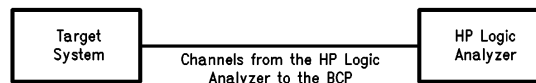

FIGURE 1. Block Diagram of the System Set Up

TL/F/10814–1

CONFIGURING THE HP LOGIC ANALYZER

The DP8344 Inverse Assembler software contains a State/Timing configuration which one may use without any changes. The designer can change this default configuration, or define an entirely new configuration to meet their own systems needs. However, certain parameters must exist in the configuration for the DP8344 Inverse Assembler to work. These parameters will be described using the default configuration as an example.

Internal communication variables are set as the logic analyzer collects data from the target system. Therefore, the logic analyzer's configuration must follow the setup described here. *Figures 4–6* show the configuration provided on the DP8344 BCP Inverse Assembler diskette. One may create their own configuration by adding more labels and connecting more channels to the target system than shown in the examples in this document. This will allow one to monitor the system activity according to their needs. However, the logic analyzers system configuration must include the following:

In the Configuration Menu, as in *Figure 4,* one must:

1. Define the Analyzer Type to be a State Analyzer.
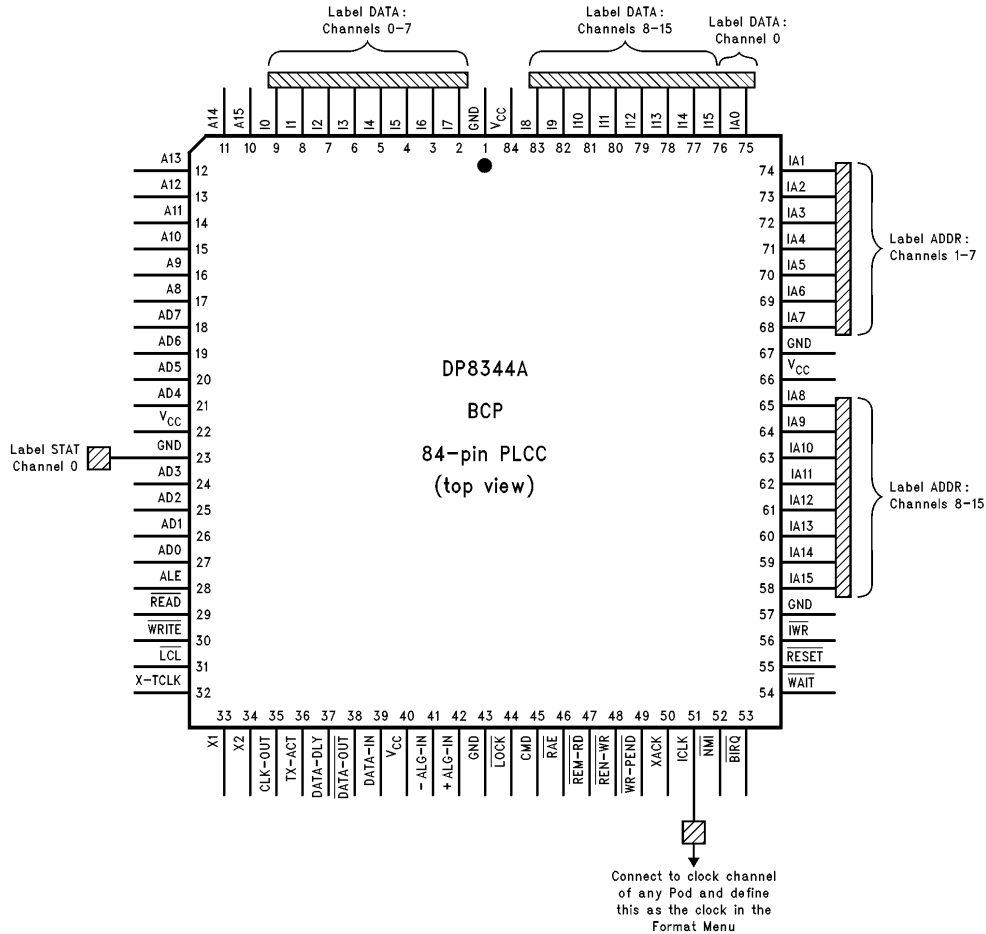
2. Assign at least two pods to the State Analyzer.

TL/F/10814–2

FIGURE 2. Pins Connected to Logic Analyzer Pods

2

| System | Front Disc | | Cancel |

| Load | All | from file | BCP |
| | | | Execute |

TL/F/10814–3

Filename	File Type	File Description
BCP	inverse__assem	DP8344 BCP INVERSE ASSEMBLER
BCP__D	16530A__config	DP8344 BCP INVERSE ASSEMBLER
BCP__E	16510A__config	DP8344 BCP INVERSE ASSEMBLER
BCP__	16500A__config	DP8344 BCP INVERSE ASSEMBLER

(a)

| System | Front Disc | | Cancel |

| Load | State/Timing E | from file | BCP_E |
| | | | Execute |

TL/F/10814–4

Filename	File Type	File Description
BCP	inverse__assem	DP8344 BCP INVERSE ASSEMBLER
BCP__D	16530A__config	DP8344 BCP INVERSE ASSEMBLER
BCP__E	16510A__config	DP8344 BCP INVERSE ASSEMBLER
BCP__	16500A__config	DP8344 BCP INVERSE ASSEMBLER

(b)

**FIGURE 3. Two Methods to Load DP8344 BCP Inverse
Assembler Software from the Front Disk Menu**

TL/F/10814–5

FIGURE 4. Configuration Menu on Logic Analyzer

In the Format Menu, see *Figure 5*, define the labels and assign the channels in the following manner:

1. Create labels ADDR, DATA, and STAT.

2. Assign the channels connected to the labels as follows:

    i. Label ADDR refers the channels connected to the In- struction memory Address Bus on the DP8344. From *Figure 2*, these are pins 75 through 68, and pins 65 through 58. To use the default configuration the pins from the Instruction memory Address bus must be connected to channel 0 through 15 of Pod E1.

    ii. The DATA label refers to the channels connected to the Instruction memory data bus on the DP8344. From *Figure 2*, these are pins 9–2, and pins 83–76. To use the default configuration the pins from the Instruction memory Data bus must be connected to channels 0 through 15 of Pod E2.

    iii. For the label STAT it is not necessary to actually con- nect any of the defined channels to the BCP. However, it is recommended that one does connect all defined channels to a pin such as ground. This is because the BCP does not use a STATUS bus. The STAT label **must** be defined in the Format Menu. In the example shown in *Figure 5*, the channel assigned to the STAT label corresponds to a ground pin on the BCP connect- ed to channel 0 of Pod E3.

3. Define the Clock to be the channel which corresponds to the connection from the pod clock connection to pin 51, ICLK, on the DP8344. In the example shown in *Figure 5,* the J clock means that ICLK is connected to the clock channel of pod E1. Set the clock to trigger on the rising edge of ICLK.
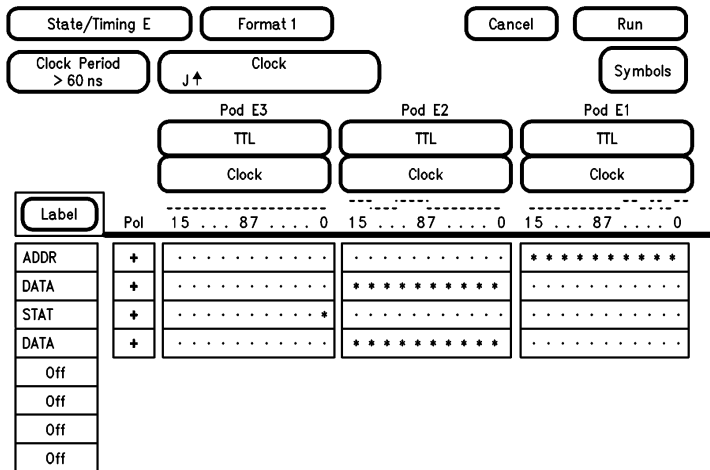

TL/F/10814–6

FIGURE 5. Format Menu on Logic Analyzer

TL/F/10814–7

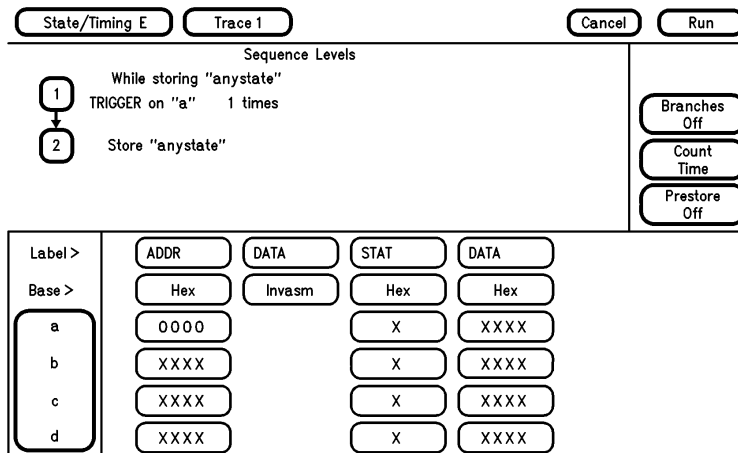**FIGURE 6. Trace Menu on Logic Analyzer**

The trigger may be defined in the Trace Menu according to the information desired. For example, in *Figure 6*, the trace is set to trigger when the BCP executes the program, i.e., the Instruction memory Address bus is 0 hex.

Once the system configuration has been developed, it must be linked with the inverse assembler software. First, load the DP8344 BCP Inverse Assembler Software by either method shown in *Figure 3*. Second, create a configuration by either:

 i. modifying the configuration file which was loaded into the HP Logic Analyzer with the DP8344 BCP Inverse Assembler, or

ii. by loading another State/Timing Configuration which has been stored on diskette.

Third, verify that the three labels: ADDR, DATA and STAT exist in the Format Menu. Fourth, in the State Listing Display, shown in *Figure 7*, select the base field below the label DATA. This will generate seven pop-outs. Select the "Invasm" pop-out to allow the mnemonics to be displayed. Finally, store the new configuration to the DP8344 BCP Inverse Assembler using one of the two methods shown in *Figure 8*. Whenever this configuration file is loaded, the inverse assembler will automatically load. Note that storing the configuration to the Inverse Assembler will write over any previously stored configurations. Therefore, it is recommended that one back up all of the stored configurations by copying them to a backup diskette.

The system is now set to capture the BCP op-codes from your system and display them as mnemonics.

| State/Timing E | Listing 1 | | Invasm | Cancel | Run |

| Markers Off |

| Label > | ADDR | DATA | DP8344 BCP MNEMONIC | Time |
| Base > | Hex | Hex | Hex | Relative |

TL/F/10814–8

FIGURE 7. State Listing Display

The Data Label with base Hex will display the op-codes in Hex Format. The DP8344 BCP MNEMONIC Label is generated by selecting the base type for the Label DATA to be "Invasm".

| System | Front Disc | | Cancel |

| Store | All | to file | BCP |

| file description : | DP8344 BCP INVERSE ASSEMBLER | Execute |

TL/F/10814–9

Filename	File Type	File Description
BCP	inverse__assem	DP8344 BCP INVERSE ASSEMBLER
BCP__D	16530A__config	DP8344 BCP INVERSE ASSEMBLER
BCP__E	16510A__config	DP8344 BCP INVERSE ASSEMBLER
BCP__	16500A__config	DP8344 BCP INVERSE ASSEMBLER

(a)

| System | Front Disc | | Cancel |

| Store | State/Timing E | to file | BCP_E |

| file description : | DP8344 BCP INVERSE ASSEMBLER | Execute |

TL/F/10814–10

Filename	File Type	File Description
BCP	inverse__assem	DP8344 BCP INVERSE ASSEMBLER
BCP__D	16530A__config	DP8344 BCP INVERSE ASSEMBLER
BCP__E	16510A__config	DP8344 BCP INVERSE ASSEMBLER
BCP__	16500A__config	DP8344 BCP INVERSE ASSEMBLER

(b)

**FIGURE 8. Two Methods to Store Configurations to the
DP8344 BCP Inverse Assembler Software**

DP8344 BCP INVERSE ASSEMBLER OPERATION

An inverse assembler converts instructions captured by the logic analyzer in binary form into mnemonics. Thus it makes it much easier to follow the program's execution flow. Furthermore, one can still use the logic analyzer to view other useful information by specifying the trace conditions, labels and channel connections in the logic analyzer's configuration file.

One needs to be aware of how the captured information is actually diassembled. The inverse assembler begins disassembling at the event which was triggered upon. Hence, any information captured prior to the trigger may not be correctly disassembled. To ensure valid disassembly of states captured prior to the trigger, one must scroll the display so the first instruction one wants disassembled is the first line on the display. Then select the "Invasm" pop-up on the top line of the State Listing Display. This causes the inverse assembler to disassemble the code from the first line on the display. For an example, refer to *Figures 9* through *12*. The inverse assembler was set to trigger when the Instruction Address Bus was 80 hex, as in *Figures 9* and *10*. The two byte instructions captured prior to the trigger were not correctly disassembled. Referring to *Figure 11*, one observes that line −10 is disassembled as an ADD Instruction rather than as the second byte of the LJMP instruction from line −11. To correct this, one must select "Invasm" from the top line of the State Listing Menu. The inverse assembler immediately disassembles the code from the first line on the screen. The correctly disassembled code is shown in *Figure 12*.

TL/F/10814–11

FIGURE 9. Triggering Event

TL/F/10814–12

	ADDR	DATA	DP8344 BCP MNEMONIC	Time
Base >	Hex	Hex	Hex	Relative
−7	005C	DD01	JMP RE, S, 005EH	120 ns
−6	005D	D501	JMP RE, NS, 005FH	80 ns
−5	005F	CF40	ILLEGAL OPCODE	160 ns
−4	0060	CF00	ILLEGAL OPCODE	120 ns
−3	0061	4FE2	AND FEH, ICR/ATR	120 ns
−2	0062	C508	ILLEGAL OPCODE	80 ns
−1	0063	CC1C	CALL 0080H	120 ns
0	0080	AE80	EXX MA, MB, NCHG	160 ns
1	0081	C343	MOVE ACR/FBR, [IV+]	80 ns
2	0082	AE90	EXX AA, MB, NCHG	160 ns
3	0083	AEC8	EXX MA, AB, EI	120 ns
4	0084	C343	MOVE ACR/FBR, [IV+]	120 ns
5	0085	AEE0	EXX MA, BM, DI	160 ns
6	0104	AFF0	RET DI, RFB	80 ns
7	0085	AEE0	EXX MA, MB, DI	120 ns
8	0086	C343	MOVE ACR/FBR, [IV+]	120 ns

FIGURE 10. Triggered Event as Shown in the State Listing

| State/Timing E | Listing 1 | Invasm | Cancel | Run |

| Markers Off |

TL/F/10814–13

Label >	ADDR	DATA	DP8344 BCP MNEMONIC	Time
Base >	Hex	Hex	Hex	Relative
−19	0050	0051	ADD 05H, NCF/IBR	120 ns
−18	0051	4009	AND 00H, GP5/GP5'	80 ns
−17	0052	C340	MOVE CCR/DCR, [IY+]	120 ns
−16	0053	CD89	JMP GP5/GP5'	160 ns
−15	0054	B00D	MOVE 00H, IWHI	200 ns
−14	0055	B57C	MOVE 57H, IWLO	120 ns
−13	0056	CD00	LJMP [IW]	120 ns
−12	0057	8009	JRMK GP5/GP5', 0H, 0H	80 ns
−11	0058	8D09	LJMP GP5/GP5', 0H, S, 0058H	200 ns
−10	0059	0058	ADD 05H, GP4/GP4'	120 ns
−9	005A	8C09	LJMP GP5/GP5', 0H, NS, 005CH	120 ns
−8	005B	005C	ADD 05H, IWLO	80 ns
−7	005C	DD01	JMP RE, S, 005EH	120 ns
−6	005D	D501	JMP RE, NS, 005FH	120 ns
−5	005F	CF40	ILLEGAL OPCODE	160 ns
−4	0060	CF00	ILLEGAL OPCODE	80 ns

FIGURE 11. Incorrectly Disassembled Instructions Prior to Triggered Event

| State/Timing E | Listing 1 | Invasm | Cancel | Run |

| Markers Off |

TL/F/10814–14

Label >	ADDR	DATA	DP8344 BCP MNEMONIC	Time
Base >	Hex	Hex	Hex	Relative
−19	0050	0051	ADD 05H, NCF/IBR	120 ns
−18	0051	4009	AND 00H, GP5/GP5'	80 ns
−17	0052	C340	MOVE CCR/DCR, [IY+]	120 ns
−16	0053	CD89	JMP GP5/GP5'	160 ns
−15	0054	B00D	MOVE 00H, IWHI	200 ns
−14	0055	B57C	MOVE 57H, IWLO	120 ns
−13	0056	CD00	LJMP [IW]	120 ns
−12	0057	8009	JRMK GP5/GP5', 0H, 0H	80 ns
−11	0058	8D09	LJMP GP5/GP5', 0H, S, 0058H	200 ns
−10	0059	0058		120 ns
−9	005A	8C09	LJMP GP5/GP5', 0H, NS, 005CH	120 ns
−8	005B	005C		80 ns
−7	005C	DD01	JMP RE, S, 005EH	120 ns
−6	005D	D501	JMP RE, NS, 005FH	120 ns
−5	005F	CF40	ILLEGAL OPCODE	160 ns
−4	0060	CF00	ILLEGAL OPCODE	80 ns

FIGURE 12. Instructions Prior to Triggered Event, Correctly Disassembled after Choosing the "Invasm" Pop-Out

This same technique must be applied if one jumps ahead in the display and then scrolls backwards to view a certain state; in other words, you do not scroll forward through every line to reach the desired state. For example, if one manually selected the line number −12 in *Figure 12* and entered line 226, the screen would display lines 219 through 234. Now if one rolls the screen backwards to display lines 199 through 214 as in *Figure 13*, the two byte instruction, LJMP, is once again not correctly disassembled. Therefore, select the ''Invasm'' pop-out and the display is correctly disassembled as shown in *Figure 14*.

One of the features of the BCP is that it uses register banks. However, there is no external indication of the bank's state. The name of a register therefore depends upon which bank one is in, as in *Figure 15*. Due to the manner in which the inverse assembler disassembles the captured data, keeping track of the correct register name meant that one would constantly have to scroll the screen back to the last EXX statement and hit the ''Invasm'' pop-out to ensure that the displayed register names are correct. Hence, to avoid this inconvenience, the register names for both banks are displayed at all times. Refer to line 45 of *Figure 16* for an example. The op-code decodes to MOVE where the source register is R0. Therefore, the register names for R0 in both banks: Main Bank A — CCR, and Alternate Bank A — DCR, are displayed.

To view the op-code in both mnemonic form and hex form, as in *Figure 16*, define the DATA label twice in the Format Menu, as in *Figure 4*. Then, select the base label to be ''Hex'' for one and ''Invasm'' for the other in the State Listing.

OBTAINING THE DP8344 BCP INVERSE ASSEMBLER

The DP8344 BCP Inverse Assembler package for use in a Hewlett Packard Logic Analyzer can be obtained from National Semiconductor. Included in the Inverse Assembler Package is the DP8344 BCP Inverse Assembler software, including configuration files as described in this application note. These will be on a 3½″ diskette formatted for use in the HP Logic Analyzer. Additionally, a 5¼″ diskette formatted for use on an IBM personal computer or compatible, containing the DP8344 Inverse Assembler source code can be obtained upon a request from National Semiconductor.

If one owns the HP 10391A Inverse Assembler Development Package, the source code can be modified to make any improvements one wishes to make to the DP8344 BCP Inverse Assembler. Note that it is not necessary to have the HP 10391A Inverse Assembler Development Package to use the DP8344 BCP Inverse Assembler.

| State/Timing E | Listing 1 | Invasm | Cancel | Run |

| Markers Off |

| Label > | ADDR | DATA | DP8344 BCP MNEMONIC | Time |
| Base > | Hex | Hex | Hex | Relative |

TL/F/10814–15

199	006A	FD08	MOVE GP4/GP4′, GP4/GP4′	120 ns
200	006B	CE00	LJMP 006AH	80 ns
201	006C	006A	ADD 06H, GP6/GP6′	120 ns
202	006A	FD08	MOVE GP4/GP4′, GP4/GP4′	120 ns
203	006B	CE00	LJMP 006AH	80 ns
204	006C	006A	ADD 06H, GP6/GP6′	120 ns
205	006A	FD08	MOVE GP4/GP4′, GP4/GP4′	80 ns
206	006B	CE00	LJMP 006AH	120 ns
207	006C	006A	ADD 06H, GP6/GP6′	120 ns
208	006A	FD08	MOVE GP4/GP4′, GP4/GP4′	80 ns
209	006B	CE00	LJMP 006AH	120 ns
210	006C	006A	ADD 06H, GP6/GP6′	120 ns
211	006A	FD08	MOVE GP4/GP4′, GP4/GP4′	80 ns
212	006B	CE00	LJMP 006AH	120 ns
213	006C	006A	ADD 06H, GP6/GP6′	120 ns
214	006A	FD08	MOVE GP4/GP4′, GP4/GP4′	80 ns

FIGURE 13. Incorrectly Disassembled Instructions Produced by Jumping Ahead in Display

Label >	ADDR	DATA	DP8344 BCP MNEMONIC	Time
Base >	Hex	Hex	Hex	Relative
199	006A	FD08	MOVE GP4/GP4′, GP4/GP4′	120 ns
200	006B	CE00	LJMP 006AH	80 ns
201	006C	006A		120 ns
202	006A	FD08	MOVE GP4/GP4′, GP4/GP4′	120 ns
203	006B	CE00	LJMP 006AH	80 ns
204	006C	006A		120 ns
205	006A	FD08	MOVE GP4/GP4′, GP4/GP4′	80 ns
206	006B	CE00	LJMP 006AH	120 ns
207	006C	006A		120 ns
208	006A	FD08	MOVE GP4/GP4′, GP4/GP4′	80 ns
209	006B	CE00	LJMP 006AH	120 ns
210	006C	006A		120 ns
211	006A	FD08	MOVE GP4/GP4′, GP4/GP4′	80 ns
212	006B	CE00	LJMP 006AH	120 ns
213	006C	006A		120 ns
214	006A	FD08	MOVE GP4/GP4′, GP4/GP4′	80 ns

FIGURE 14. Instructions from *Figure 13* **Correctly Disassembled after Choosing the "Invasm" Pop-Out**

Alternate — Main

A:
DCR	CCR	R0
IBR	NCF	R1
ATR	ICR	R2
FBR	ACR	R3

B:
RTR	GP0	R4
TSR	GP1	R5
TCR	GP2	R6
TMR	GP3	R7
GP4′	GP4 (accumulator)	R8
GP5′	GP5	R9
GP6′	GP6	R10
GP7′	GP7	R11

Index Registers (pointers)
W (low byte)	R12
W (high byte)	R13
X (low byte)	R14
X (high byte)	R15
Y (low byte)	R16
Y (high byte)	R17
Z (low byte)	R18
Z (high byte)	R19

GP8	R20
GP9	R21
GP10	R22
GP11	R23
GP12	R24
GP13	R25
GP14	R26
GP15	R27

Timer
TRL	R28
TRH	R29

Stacks
ISP	R30
DS	R31

FIGURE 15. Register Map

| State/Timing E | Listing 1 | Invasm | Cancel | Run |

| Markers Off |

| Label > | ADDR | DATA | DP8344 BCP MNEMONIC | Time |
| Base > | Hex | Hex | Hex | Relative |

| | | | | | TL/F/10814–17 |

	45	002B	C340	MOVE CCR/DCR, [IY+]	160 ns
	46	002C	E968	SUBA GP4/GP4′, GP7/GP7′	160 ns
	47	002D	C340	MOVE CCR/DCR, [IY+]	120 ns
	48	002E	A52B	SUBA GP7/GP7′, [IX+]	160 ns
	49	002F	C340	MOVE CCR/DCR, [IX+]	160 ns
	50	0030	207B	SUB 07H, GP7/GP7′	160 ns
	51	0031	C340	MOVE CCR/DCR [IY+]	80 ns
52	0032	ED6B	SBCA GP7/GP7′, GP7/GP7′	160 ns
	53	0033	C340	MOVE CCR/DCR [IY+]	120 ns
	54	0034	A72B	SBCA GP7/GP7′ [IX+]	160 ns
	55	0035	C340	MOVE CCR/DCR [IY+]	160 ns
	56	0036	A92B	ANDA GP7/GP7′, [IX+]	160 ns
	57	0037	C340	MOVE CCR/DCR, [IY+]	160 ns
	58	0038	F573	ORA IZHI, GP7/GP7′	160 ns
	59	0039	C340	MOVE CCR/DCR, [IY+]	80 ns
	60	003A	AB2B	ORA GP7/GP7′, [IX+]	160 ns

**FIGURE 16. Listing of Inverse Assembler on Logic Analyzer
Demonstrating the Display of Both Register Bank Names**

APPENDIX A

Flow Chart of DP8344 Inverse Assembler Source Code
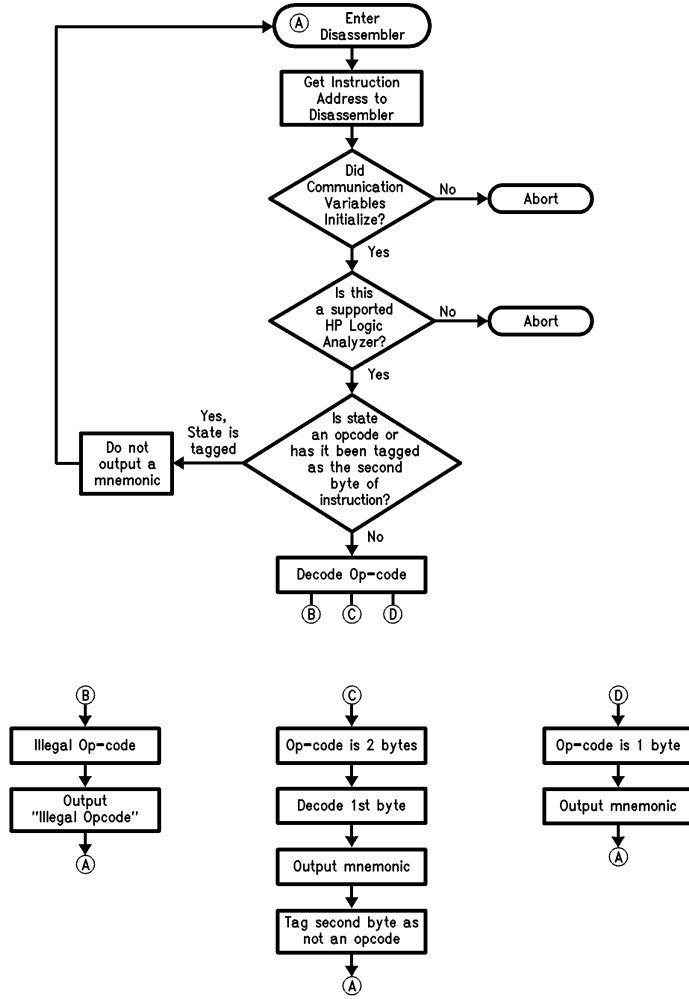
TL/F/10814–18

Lit. # 100688

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.

2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied and National reserves the right at any time without notice to change said circuitry and specifications.