H8/330

Application Note

Parallel-to-Parallel Print Buffer Controller

Tom Hampton

INTRODUCTION

The HD6473308 (H8/330) is a highly integrated 8-bit microcomputing unit. Along with a central processing unit utilizing a reduced instruction set designed for speed, the H8/330 incorporates several system peripheral devices and memory onto a single chip. These on-chip functions include 16K bytes of ROM or EPROM, 512 bytes of RAM, 15 bytes of dual-port RAM, 5 timers, a UART channel, 8 channels of A/D conversion, and 9 I/O Ports.

These features allow the H8/330 to be used in many applications; a print buffer is merely one of the vast possibilities. In this application, we are able to examine the usage of several of the on-chip peripherals as well as I/O ports and interrupt control. While this application does not use all of the peripheral features of the H8/330, it does provide programming examples for many of the peripherals as well as the CPU itself.

Three of the on-chip timers are used to control such events as strobe generation for both output data and printer initialization, and also for event monitoring. This is accomplished through the exception processing features of the H8/330. Four external interrupts are utilized to monitor input data strobes, input initialization strobes, and pushbutton (control panel) events.

In order to maximize the external memory addressing capabilities for this application, the H8/330 device is used in the single-chip mode of operation. In this mode, many of the I/O ports are used to control the memory buffer itself as well as for status displays. One of the I/O ports is even used for a bidirectional data bus even though the H8/330 does not have that feature directly.

Even though this application uses very little on-chip memory (less than 512 bytes of ROM and less than 20 bytes of RAM), the on-chip memory capabilities of the H8/330 provide enough room for code and data storage required by most applications.

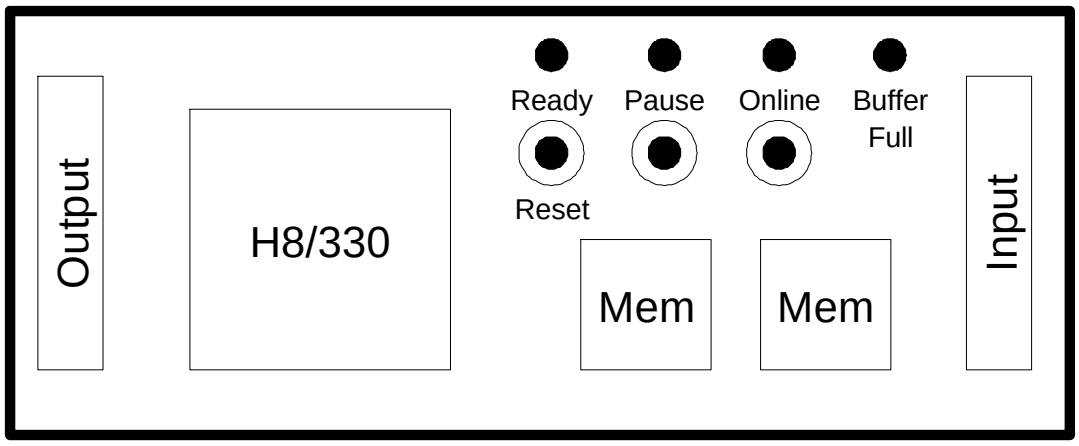

Figure 1: Buffer Block Diagram

DESIGN CONSIDERATIONS

One of the main considerations in the demo design was to keep the parts count to a minimum. This meant that we would have to use the features of the H8/330 wherever possible rather than an external device. A simple block diagram of the print buffer is shown in Figure 1.

The required parts had to consist of the H8/330 and some memory chips. We chose to use two (2) HM62256 SRAMs (32Kx8) to provide us with a 64K byte buffer. In order to clean up the power-on reset circuitry, we chose to add a 74HC14 although it probably wasn't necessary. This kept our parts count to only four ICs.

We also wanted to have some control over the operation of this print buffer. For this reason, three (3) switches were added to provide for an external reset, a means of taking the buffer "off-line" (just as if it were a printer), and a means of halting the buffer's output. We also wanted to have an indication of this control, so four (4) LEDs were added to indicate the status of the buffer.

You will find a complete schematic of the H8/330 Print Buffer in Appendix A (Figure A-1). You may want to refer to this schematic as we discuss our decisions for devices and I/O port usage.

I/O PORT USAGE

In the expanded modes of operation, the H8/330 has the capability of directly addressing external memory through the use of twenty-seven (27) of its I/O lines. We could have used one of these modes of operation, but that would limit the size of our storage buffer to considerably less than 64K bytes. Also, in these modes of operation, the only two ports on the H8/330 that have the capability to drive LEDs directly also serve as the external address bus. In order not to require the use of an external device to drive LEDs, and also to allow a large storage buffer (we chose 64K bytes for simplicity), the single-chip mode of operation was used. This forces us to use individual I/O ports to control buffers addressing, memory control, and data access. We are, however, not losing the use of any I/O lines because of this.

STATUS DISPLAY

In this design, we have four (4) LEDs that are used to display the status of the print buffer. These status indicators include Ready, Online, Buffer Full, and Output Hold. Since only ports 1 and 2 have the capability of driving LEDs directly, neither could be used to address the external memory buffer. Port 1 was chosen to indicate the status.

CONTROL PANEL

Also in this design, we have two (2) switches which are used to provide user control over the actions of the print buffer. These two switches allow the user to halt (or restart) the buffer's output, and take the print buffer on-line or off-line. Since continual polling of these switches would take too much time out of the spooling action, it was decided to use external interrupts as the switch inputs. This meant that Port 9 would be used for this function. I was also convenient since Port 9 has internal MOS pull-up resistors, thus keeping with our constraint of minimizing the parts counts.

The third switch of the control panel controls a hardware reset to the print buffer in the event that the user wishes to reset the buffer during its normal operation.

PARALLEL INPUT AND OUTPUT

Since this print buffer is parallel-in and parallel-out, three (3) 8-bit ports are required to allow for this interface (data plus handshake). The A/D converter of the H8/330 is not being utilized for this application so I/O Port 7 is ideal for the parallel input port (since it happens to be an input only port anyway). Also, since no other external interrupts are required and the free-running timer interrupt is internal only, I/O Port 6 is an ideal selection for the parallel output port.

Additionally, three control signals ($\overline{\text{INIT}}$, $\overline{\text{STB}}$, and BUSY) from both the input and output ports are necessary for proper operation.

The INIT strobe ($\overline{\text{IINIT}}$) from the input port is fed directly to the NMI input of the H8/330. When the personal computer generates this strobe, it is an indication that the system hardware wishes to reset the printer. For this reason, this event takes precedence over all others. When this occurs, the print buffer will generate an INIT pulse ($\overline{\text{OINIT}}$) for the output port to reset the printer. This pulse is also generated during the initialization sequence of the print buffer.

The input STB signal ($\overline{\text{ISTB}}$) is accepted as a maskable interrupt to Port 9. This strobe has the lowest priority of all maskable external interrupts in order to ensure that the initialization pulse and the switch press interrupts take precedence. This event causes the generation of the input BUSY signal (IBUSY) so that no other input $\overline{\text{STB}}$s can occur until the data is properly buffered.

Three of the on-chip timers are used for strobe generation and event monitoring. One of the 8-bit multi-function timers is used to generate an output data strobe while the second multi-function timer is used to generate an output printer initialization strobe. Since the multi-function timers can control both of these outputs directly, Port 4 was used for this function. The 16-bit free-running timer is used for event monitoring; its function is basically that of a watchdog timer. This timer is started and allowed to run continuously until an input service is requested. During the service of the input interrupt request, this timer is reset. If the timer is allowed to overflow, then the software assumes that there was no input activity and will check to determine if any output service can be performed. This timer is also reset during the output service routine. The interrupt generated by this timer is used internally only and not brought out to the rest of the system.

MEMORY BUFFER

The interface to the memory buffer requires fifteen (15) lines for address (32K), eight (8) lines for data access, two (2) lines for chip selection, and one (1) line for write control. Since the memory devices draw the most power when they are chip selected, it was unnecessary to use an I/O line to control the output enables. These lines were left tied to ground so that they would always be active. In this manner, the chip select (CS_n) signal activate the appropriate memory device and keep power consumption to a minimum. I/O Port 5 happens to be a 3-bit port and is ideal for use for the three control signals.

Since Port 8 is already a 7-bit port, it was convenient to use it for the most significant portion (MA_{14}-MA_8) of the address bus. Two more 8-bit ports are required to complete the address bus as well as the data access bus. Since only two full 8-bit ports remained, we used Port 2 for the data bus (MD_7-MD_0) and Port 3 for the least significant byte (MA_7-MA_0) of the address bus.

SOFTWARE

The print buffer software basically consists of eight separate routines.

1. The main routine performs initialization of the I/O ports, timers, and the memory buffer as well as the generation of an initialization pulse for the printer attached to the buffer.

2. The input strobe (ISTB) service routine is responsible for placing data from the input data port into the memory buffer. This routine has the lowest priority of all the external interrupt routines, but does take precedence over all internal interrupts.

3. The output service routine is responsible for placing data from the memory buffer out to the output port. This routine is allowed to occur only when no other higher priority activity ($\overline{\text{IINIT}}$, $\overline{\text{ISTB}}$, or control panel switch press) has requested service within 100 µsec. This routine is also responsible for generating the output data strobe ($\overline{\text{OSTB}}$).

4. The output strobe service routine takes care of disabling the multi-function timer from generating further output strobes ($\overline{\text{OSTB}}$) as requested by the output service routine.

5. The input initialization ($\overline{\text{IINIT}}$) service routine provides a means by which the sending device can reset the print buffer as well as the printer connected to it. This includes generation of an output initialization strobe ($\overline{\text{OINIT}}$) as well as initializing the print buffer

6. The initialization strobe routine takes care of disabling the multi-function timer from generating further output initialization strobes ($\overline{\text{OINIT}}$) as requested by the input initialization service routine.

7. The online pushbutton service routine monitors an external switch which allows the user control over whether or not to allow data to be input to the print buffer. This performs a function similar to a printer's "online" switch.

8. The pause pushbutton service routine also monitor an external switch. This routine allows the user control over allowing the data in the memory buffer to be output to the printer. This would be useful in instances where the printer's "online" switch might not be readily accessible.

Complete source listings for each routine, as well as supporting files, can be foundin Appendix A.

MAIN ROUTINE

This routine performs the function of initializing the print buffer for operation. For a flow chart of its sequence, please refer to Figure 2. In order to prevent any interrupts from being requested by the input port, the main routine sets the IBUSY signal active. By setting this signal active, the sending device is inhibited from generating any strobe ($\overline{\text{ISTB}}$) signals to the print buffer. Since the ISTB interrupt is the only interrupt that

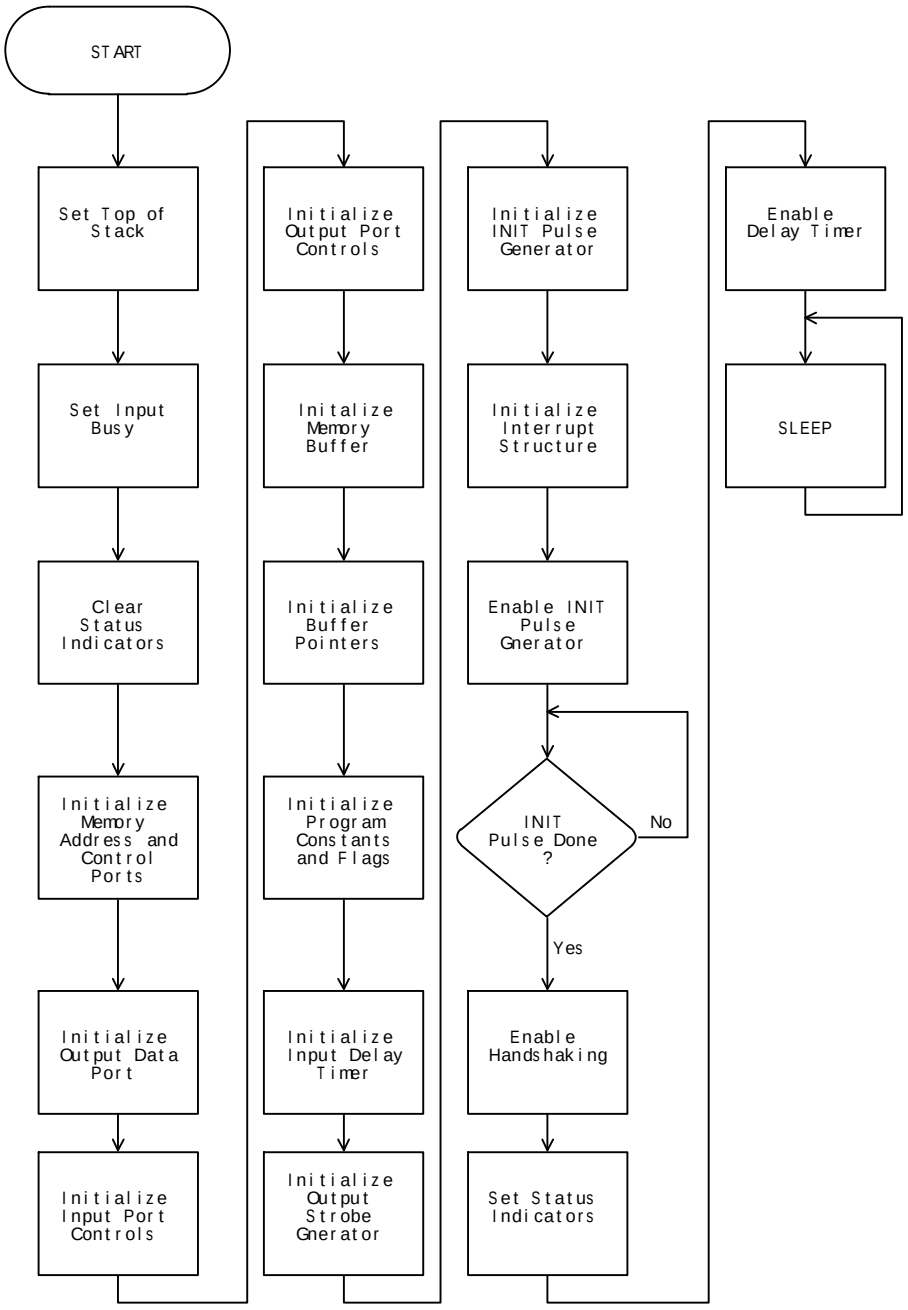

Figure 2: Main Routine

could possibly be requested before the H8/330 is initialized [except for NMI (IINIT), which performs the same function as a reset], we can now go about the business of initializing the H8/330 without worrying about missing a request for buffering.

At this point, we go through the process of initializing all of the I/O ports for proper usage. Since all of the I/O ports of the H8/330 are initialized as input ports at reset, we must go through each port and setup both direction and functions.

I/O Port 1 is used for two functions; status indication and input busy (IBUSY) signal control. All of these signals are outputs. The initial status for outputs on this port should be all high. This "turns" the LEDs off and sets the IBUSY signal active. To ensure that this happens as soon as the port is programmed for output function, we write to the data register prior to setting the direction.

The second set of ports we initialize are for the memory buffer. In order to make sure that the memory is inactive when we program the ports, we write to the I/O port (Port 5) used for the control signals to make the WE and CS signals inactive when the direction is changed to outputs. We are then able to program the I/O ports to be used as the address bus. Both of these ports (Port 3 and Port 8) are to be used as outputs. The only other port used in connection with the memory buffer is for data access. Since this port will be used bidirectionally, the direction will be programmed as we need to use it.

We can now program the ports that we intend to use for the parallel input and output ports. Since we have chosen Port 7 for the input port, no direction programming is necessary because Port 7 is input only anyway. Port 6 is used for the output parallel port so its direction must be changed. Port 9 is used for control over both the input and output parallel ports. The signals involved here are the input strobe (ISTB) and the two control panel switches for online and output hold. Since all three of these signals are inputs, no direction change is required. We can, however, use the internal MOS pull-up feature of the port so that external resistors are not needed. To control this feature, we can write to the port data register with "1s" to enable the pull-ups (this feature is available only with port bits that are inputs).

Port 4 is used for control signals directly affecting the output parallel port. These signals include the output strobe (OSTB), output busy (OBUSY), and output initialization (OINIT). Since the OBUSY signal is an input while the other two signals are outputs, we must program this port for a mixture of input and output lines. Initially, we would like to ensure that the OSTB signal is inactive while the OINIT is active (to make certain that the printer itself gets reset) and to enable the MOS pull-up on the OBUSY signal. To do this we program the data register of the port with the value H'13 before programming the direction of the port. For setting up bits 4 and 1 as outputs while bit 0 is an input, we must program the direction register with the value H'12.

At this point in the main routine we finally come to where we get to use the I/O ports for controlling the memory buffer. We use this opportunity to clear the memory buffer contents. In performing this operation, we must first set the I/O port used for the data bus to the output direction. We can then setup the ports used for the memory address with a valid address as well as the data port with the clearing value to be written. Next we write to the I/O port used for the control signals to activate the WE and correct CS signal. Since were are using different instructions to set and clear these bits, we can immediately deactivate the signals after having activated them. This sequence provides plenty of time for proper SRAM operation. This function is positioned inside a loop that executes until the entire buffer has been cleared.

We are now at a position where we can set the program constants and operation flags, as well as initializing the timers for their uses. The 16-bit free-running timer is used for watching input activity. This timer is programmed to generate an interrupt every 100 µsec, but this occurs only if it is not reset in a service routine. One of the 8-bit multi-function timers is used to generate the output data strobe. This timer is programmed to generate a negative-going strobe that is 2.4 µsec in width, and an interrupt on the trailing edge of that pulse. Since we don't want to generate the strobe at this time, the timer output is programmed to remain at a high level with no interrupts generated. We control the output level during the output service routine. The second 8-bit multi-function timer is used to generate the output initialization (OINIT) pulse. This timer is programmed to generate a negative-going strobe that is 40 µsec in width, and an interrupt on the trailing edge of that pulse. Again, we do not wish to generate the strobe right now so the timer output is programmed to remain at a high level with no interrupts generated. We will generate the strobe right after we initialize the interrupt structure.

The interrupt structure must be setup to handle three (3) external interrupts and also for those external interrupts generated by the falling edge of a signal. For the external interrupts, this is accomplished by programming the Interrupt Sense Control Register (ISCR) for edge selection prior to programming the Interrupt Enable Register (IER). The interrupt mask is then released in the Condition Codes Register (CCR) of the H8/330 in order to enable all interrupts.

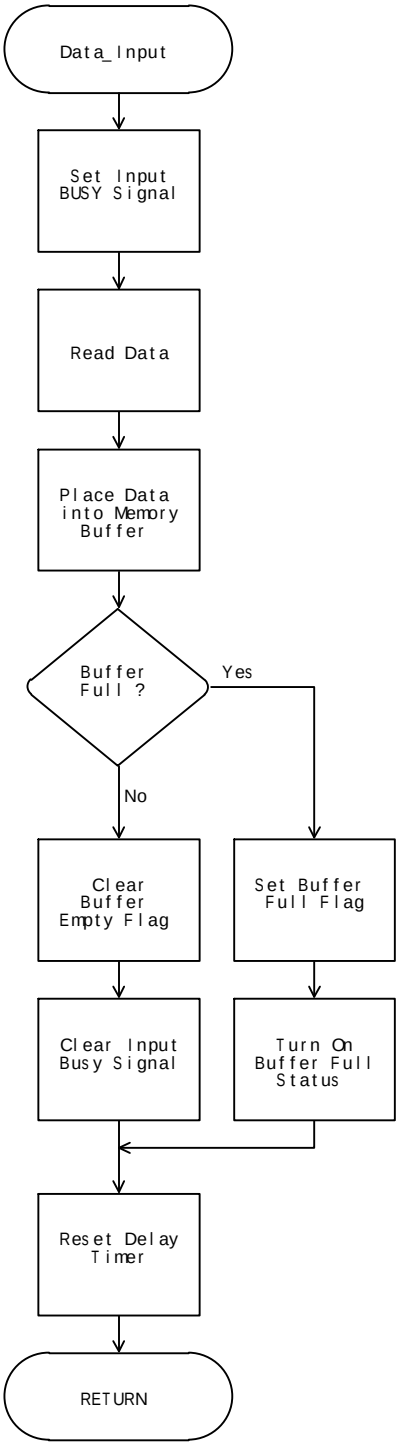

Figure 3: Data Input Service Routine

Now we can enable the second 8-bit multi-function timer to generate the $\overline{\text{OINIT}}$ signal and also the interrupts. Here we wait in a loop until the "oinit" status bit has been cleared to indicate that the pulse has occurred. Now we clear out the IBUSY signal so that input data strobes can occur, and set the status indicators to show that the print buffer is "online" and "ready."

The last thing we do in the main routine is to enable the input watchdog timer so that its interrupts can be generated.

Data Input Service Routine

The input data strobe ($\overline{\text{ISTB}}$) is input to the H8/330 as the lowest level maskable interrupt. Whenever the falling edge of the $\overline{\text{ISTB}}$ signal is detected, the H8/330 goes through the process of inputting data from the parallel port and placing into the print buffer. For a flow chart of this service routine, refer to Figure 3. In order to keep further $\overline{\text{ISTB}}$ interrupts from occurring before the print buffer is ready to accept them, this routine first sets the IBUSY signal active before it can do anything else. It then goes through the process of getting the data and writing it to the memory buffer.

A separate pointer is maintained for the input position of the buffer. This position is checked against the output data pointer to determine when the buffer is to full to accept any more data. As long as the buffer is not full, the "buffer empty" flag is cleared, the input watchdog timer is reset, and the IBUSY signal is deactivated. This would complete this service routine. If the memory buffer is determined to be full, then the "buffer full" flag is set, the "Buffer Full" status indicator is "turned on," and the service routine completes with the IBUSY signal remaining active after resetting the input watchdog timer. This inhibits further input strobe interrupts from occurring until the buffer has been emptied of some of its data.

Data Output Service Routine

One of the timers is initialized such that it will generate an interrupt to the H8/330 if no input or output activity takes place within 100 µsec. Both the data input and data output service routines reset this counter in order to keep both activities going. For a flow chart of this service routine, refer to Figure 4. To ensure that no input data requests are generated while the output service is taking place, this routine also sets the IBUSY signal active immediately.

Since a timer generated this request rather the printer itself, this routine must determine whether or not there is data in the buffer to be output and whether or not the printer is ready to

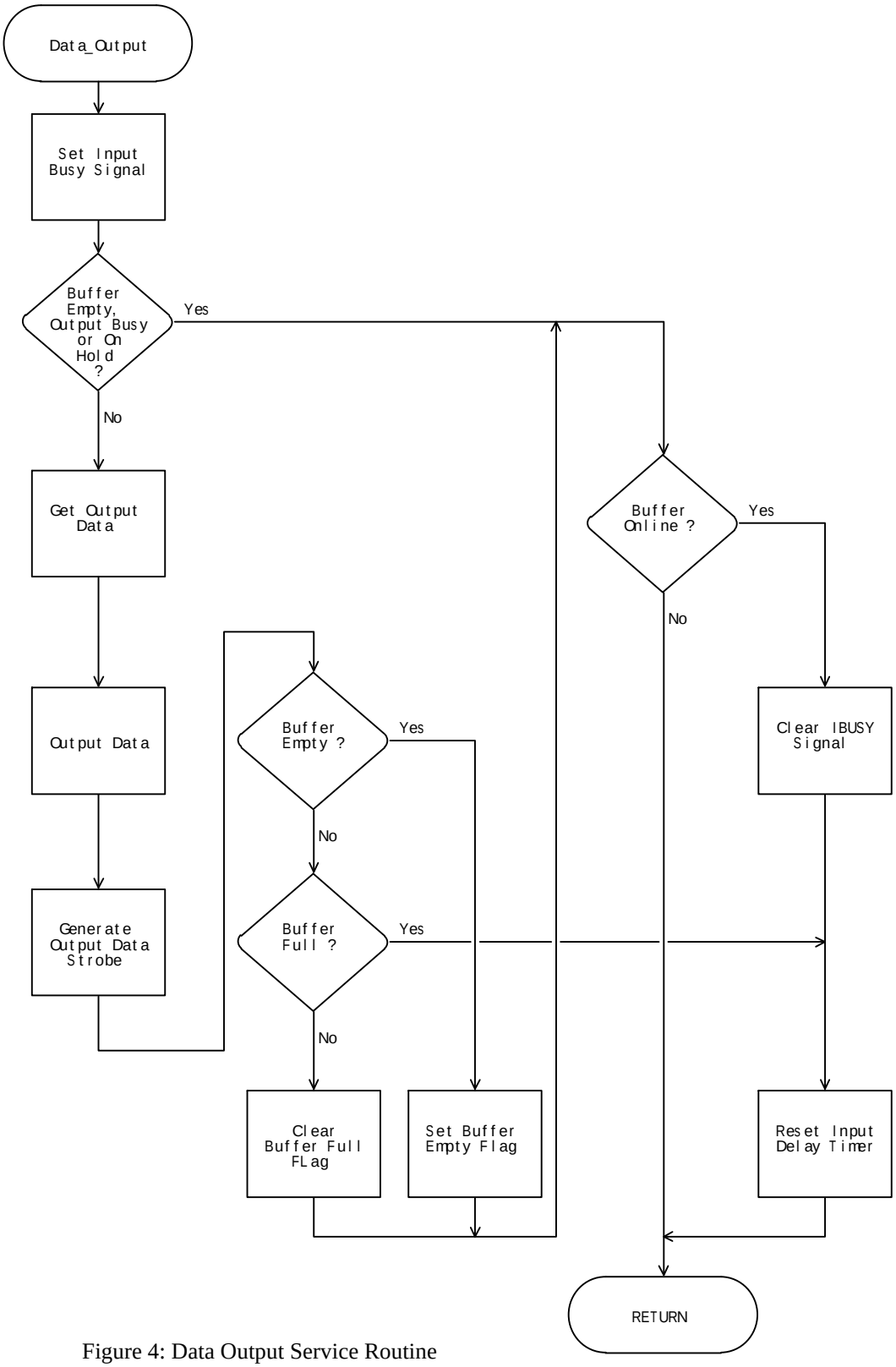

Figure 4: Data Output Service Routine

accept such data. If no data is in the buffer or the printer is not ready to accept the buffered data, this routine merely resets the input watchdog timer and checks to determine if the IBUSY signal should be activated before returning to the main program.

If there is data in the buffer and the printer is ready to accept the data, then this routine goes through the process of getting the data from the buffer and sending it to the output parallel port. In getting the data from the memory buffer, this routine must change the direction of the memory buffer's data bus to be input as well as set the address bus with the output data pointer. The proper $\overline{CS}$ signal is then generated in order to "read" the data to be output. That data is moved to the output parallel port and the multi-function timer that generates the output strobe ($\overline{OSTB}$) is enabled. The H8/330 then goes to "sleep" until the output strobe interrupt occurs.

After returning from the output strobe interrupt routine, the data output service routine has to determine whether or not the memory buffer is in a either the "empty" or "full" condition. If the buffer is in the empty condition, then this routine sets the "buffer empty" flag, deactivates the IBUSY signal, and resets the input watchdog timer in completing its operations. If the buffer is not "empty," then this routine must determine whether or not the buffer is in the "full" condition. If the buffer is in a "full" condition, then this routine just resets the input watchdog timer and completes its service with the IBUSY signal still being set. If the buffer is not in the "full" condition, then the routine clears the "buffer full" flag, deactivates the IBUSY signal, and resets the input watchdog timer in completing its operations.

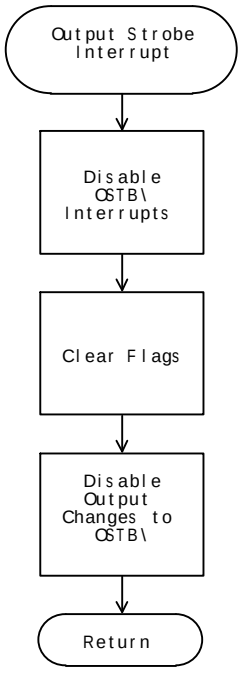

Figure 5: Output Strobe Service Routine

Output Strobe Service Routine

During the execution of the data output service routine, one of the 8-bit multi-function timers is programmed to generate the $\overline{OSTB}$ signal, and also an interrupt on the trailing edge of that strobe. During this service routine, this timer is programmed not to generate any more interrupts and also to keep its output at a high level (thus generating no more strobes). Execution then returns to the data output service routine. For a flow chart of this service routine, refer to Figure 5.

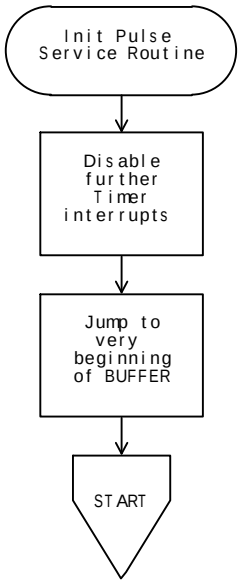

Figure 6: $\overline{IINIT}$ Pulse Service Routine

Input INIT Pulse Service Routine

The input $\overline{INIT}$ pulse ($\overline{IINIT}$) signal is connected directly to the $\overline{NMI}$ input of the H8/330. Whenever the sending device sets this signal active, the print buffer will disable all timers from generating any of their interrupts. It then restarts the software just as if the reset pushbutton had been pressed. This allows the sending device to control the reset of the buffer and printer through its hardware signal. It does not interfere with any software reset commands sent directly to the printer. For a flow chart of this service routine, refer to Figure 6.

Output INIT Pulse Service Routine

During the execution of the main routine, one of the 8-bit multi-function timers is programmed to generate the $\overline{OINIT}$ signal, and also an interrupt on the trailing edge of that strobe. During this service routine, this timer is programmed not to generate any more interrupts and also to keep its output at a high level (thus generating no more strobes). The routine also clears the "oinit" status bit so that the main routine can complete it operation. For a flow chart of this service routine, refer to Figure 7.

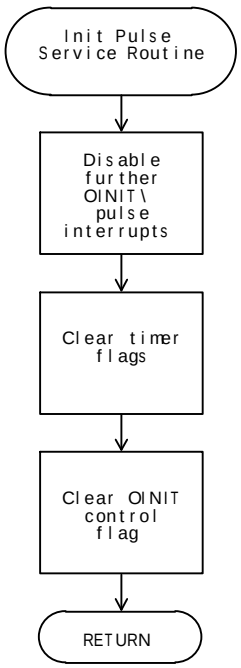

Figure 7: $\overline{OINIT}$ Pulse Service Routine

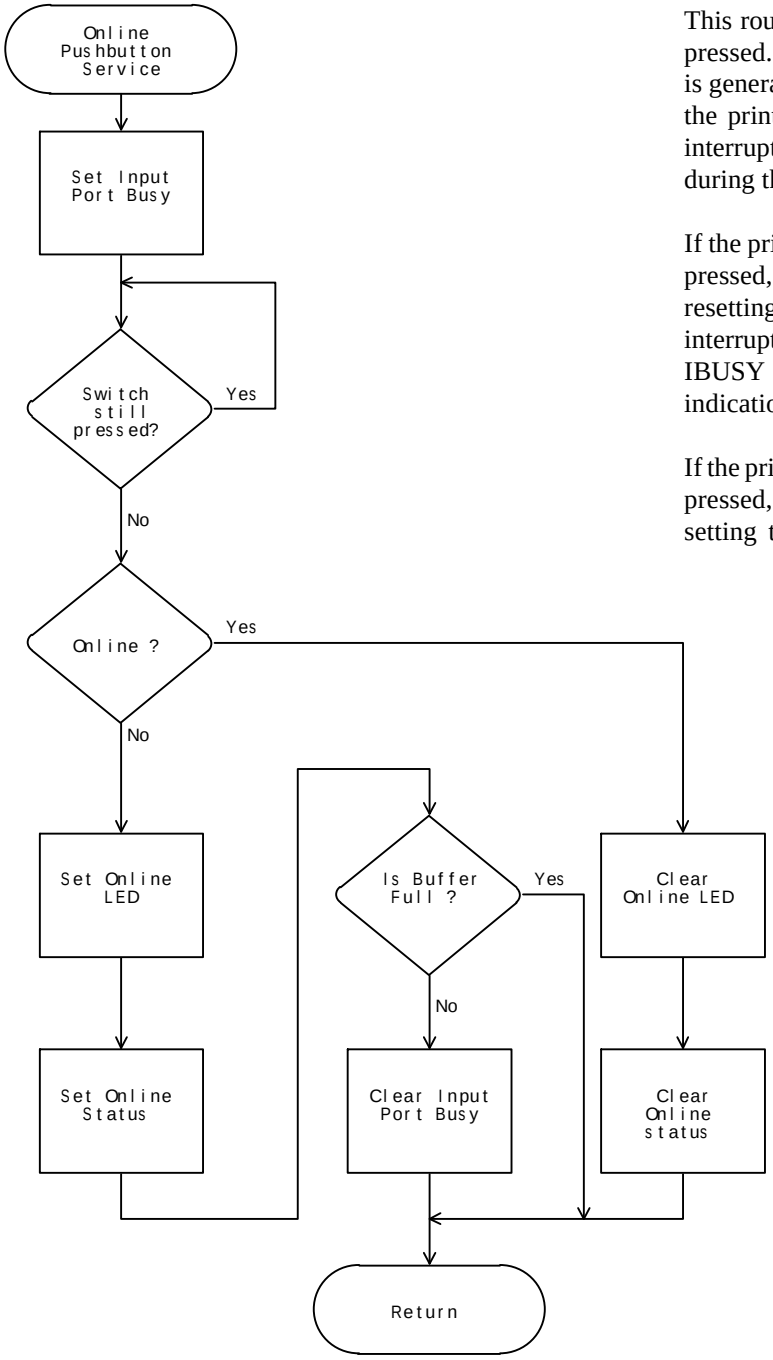

Figure 8: Online Pushbutton Service Routine

ONLINE PUSHBUTTON SERVICE ROUTINE

This routine executes whenever the "Online" pushbutton is pressed. Whenever this button is pressed, an interrupt request is generated that allows the software to control the ability of the print buffer to accept any input data. To inhibit input interrupts from being requested, the IBUSY signal is set active during the processing of this routine.

If the print buffer is currently online when this pushbutton is pressed, then this routine will take it off-line. This is done by resetting the "online" status flag and disabling input strobe interrupts. The $\overline{\text{ISTB}}$ interrupt itself is disabled as well as the IBUSY signal left active so that the sending device has an indication that the buffer is now "off-line."

If the print buffer is currently off-line when this pushbutton is pressed, then this routine will take it online. This is done by setting the "online" status flag and enabling input strobe interrupts. The $\overline{\text{ISTB}}$ interrupt itself is enabled as well as the IBUSY signal made inactive so that the sending device has an indication that the buffer is now "online." For a flow chart of this service routine, refer to Figure 8.

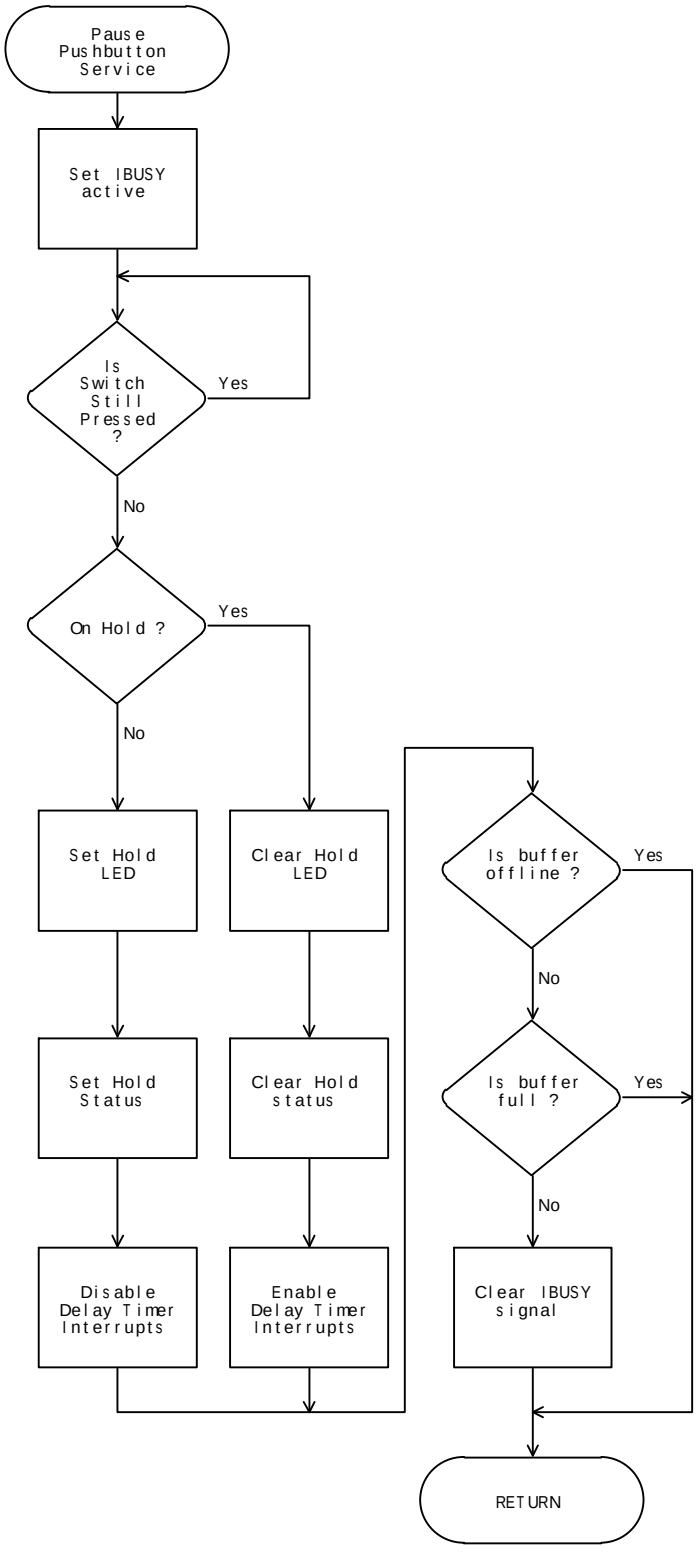

Figure 9: Pause Pushbutton Service Routine

PAUSE PUSHBUTTON SERVICE ROUTINE

This routine executes whenever the "Pause" pushbutton is pressed. Whenever this button is pressed, an interrupt request is generated that allows the software to control the ability of the print buffer to output any input data to the printer. This would be very similar to having pressed the "Online" pushbutton at the printer itself.

If the print buffer output is currently active when this pushbutton is pressed, then this routine will make it inactive. This is done by setting the "output hold" status flag and disabling input watchdog interrupts.

If the print buffer output is currently inactive when this pushbutton is pressed, then this routine will make it active. This is done by resetting the "output hold" status flag and enabling input watchdog interrupts. For a flow chart of this service routine, refer to Figure 9.

CONCLUSION

While this example does not use all of the peripheral features of the H8/330, it does provide examples of programming for both timers and I/O ports, as well as features of the individual I/O ports. Also included are methods for initializing the interrupts structure of the H8/330. Enhancements can most certainly be made to this example by doing some rearranging of the I/O port choices. A serial input or output option can be made by using the on-chip SCI and moving the memory buffer control functions to another I/O port. More memory could be added by using more I/O bits from another port to expand the address field. This would also required a little extra address manipulation in determining buffer conditions, but it is achievable.

APPENDIX A

TABLE OF CONTENTS

Figure A-1: Print Buffer Schematic Diagram
Listing 1: Vector initialization ..... BUFFER.LIS
Listing 2: Buffer initialization ..... START.LIS
Listing 3: Input strobe service routine ..... INPUT.LIS
Listing 4: Output service routine ..... OUTPUT.LIS
Listing 5: Output data strobe service routine ..... OUT-STB.LIS
Listing 6: Input initialization pulse service routine ..... IN-INIT.LIS
Listing 7: Output initialization pulse service routine ..... OUT-INIT.LIS
Listing 8: "Online" pushbutton service routine ..... ONLINE.LIS
Listing 9: "Pause" pushbutton service routine ..... PAUSE.LIS
Listing 10: Print buffer design equates ..... BUFFER.INC
Listing 11: H8/330 equates H8330.INC

paste schematic diagram here

Figure A-1: Schematic Diagram

Listing 1: BUFFER.LIS

```
*** H8/300 ASSEMBLER               VER 1.1 ***   03/20/91 08:11:13                                                PAGE     1
PROGRAM NAME =                                  Vector Table Definitions

    1                                                   .heading        “Vector Table Definitions”
    2
    3                                           ;H8/330 Print Buffer Routine
    4                                           ;revision 2.0
    5
    6                                           ;written by:
    7                                           ;       Tom Hampton
    8                                           ;       Hitachi America, Ltd.
    9                                           ;       Application Engineering
   10
  248                                                   .output         dbg,obj
  249                                                   .print          nocref,nosct
  250
  251                                                   .global         start
  252                                                   .global         online_int,pause_int,input_int,iinit_int
  253                                                   .global         output_int,ostb_int,oinit_int
  254
  255 vector   D 0000                                   .section        vector,data,locate=0
  256
  257                                           ;vector table initialization
  258
  259 vector   D 0000                                   .org    0
  260 vector   D 0000 0000                             .data.w start           ;reset vector
  261
  262 vector   D 0006                                   .org    nmi_vec
  263 vector   D 0006 0000                             .data.w iinit_int       ;input init strobe detect
  264
  265 vector   D 0008                                   .org    irq0_vec
  266 vector   D 0008 0000                             .data.w online_int      ;online pushbutton detect
  267
  268 vector   D 000A                                   .org    irq1_vec
  269 vector   D 000A 0000                             .data.w pause_int       ;pause pushbutton detect
  270
  271 vector   D 000C                                   .org    irq2_vec
  272 vector   D 000C 0000                             .data.w input_int       ;input strobe detect
  273
  274 vector   D 0020                                   .org    ocia_vec
  275 vector   D 0020 0000                             .data.w output_int      ;output service request
  276
  277 vector   D 0026                                   .org    cmi0a_vec
  278 vector   D 0026 0000                             .data.w ostb_int        ;output data strobe generator
  279
  280 vector   D 002C                                   .org    cmi1a_vec
  281 vector   D 002C 0000                             .data.w oinit_int       ;output init strobe generator
  282
  283                                                   .end
  *****TOTAL ERRORS       0
  *****TOTAL WARNINGS     0
```

Listing 2: START.LIS

```
*** H8/300 ASSEMBLER               VER 1.1 ***   03/20/91 08:11:20                                                PAGE     1
PROGRAM NAME =                                   Buffer Initialization Routine

    1                                            .heading        “Buffer Initialization Routine”
    2
    3                                       ;H8/330 Print Buffer Routine
    4                                       ;revision 2.0
    5
    6                                       ;written by:
    7                                       ;       Tom Hampton
    8                                       ;       Hitachi America, Ltd.
    9                                       ;       Application Engineering
   10
  248                                               .output         dbg,obj
  249                                               .print          nocref,nosct
  250
  251                                               .global         start
  252
  253 program  C 0000                               .section        program,code
  254
  255                                       ;initialization routines
  256 program  C 0000                       start:
  257
  258                                       ;initialize stack pointer
  259 program  C 0000 7907FF80                      mov.w   #top_ram,r7     ;set SP to top of ram
  260
  261                                       ;initialize input handshake and status indicators
  262 program  C 0004 F8FF                          mov.b   #h’ff,r0l       ;set IBUSY active to keep
  263 program  C 0006 38B2                          mov.b   r0l,@p1_dr      ; istb interrupts inactive
  264                                                                       ;clear LEDs
  265 program  C 0008 38B0                          mov.b   r0l,@p1_ddr     ;set port as outputs
  266
  267                                       ;initialize Memory Control Port
  268 program  C 000A 38BA                          mov.b   r0l,@p5_dr      ;set WE\, CS1\, & CS0\ inactive
  269 program  C 000C 38B8                          mov.b   r0l,@p5_ddr     ;set pins as outputs
  270
  271                                       ;initialize Memory Address Bus
  272 program  C 000E 38B6                          mov.b   r0l,@p3_dr
  273 program  C 0010 38BF                          mov.b   r0l,@p8_dr      ;set buffer address as FFFF
  274 program  C 0012 38B4                          mov.b   r0l,@p3_ddr
  275 program  C 0014 38BD                          mov.b   r0l,@p8_ddr     ;set ports as outputs
  276
  277                                       ;initialize Output Data Port
  278 program  C 0016 38B9                          mov.b   r0l,@p6_ddr     ;set port as output
  279
  280                                       ;initialize Input Port Controls and Pause
  281 program  C 0018 38C1                          mov.b   r0l,@p9_dr      ;turn-on MOS Pull-Ups
  282
  283                                       ;initialize Output Port Controls
  284 program  C 001A F813                          mov.b   #h’13,r0l
  285 program  C 001C 38B7                          mov.b   r0l,@p4_dr      ;set OINIT\ active and OSTB\ inactive
  286 program  C 001E F812                          mov.b   #h’12,r0l       ;set port as follows:
  287 program  C 0020 38B5                          mov.b   r0l,@p4_ddr     ; Bit 4 as output (OINIT)
  288                                                                       ; Bit 1 as output (OSTB\)
  289                                                                       ; Bit 0 as input (OBUSY), MOS Pull-Up active
  290
  291                                       ;initialize memory buffer
  292 program  C 0022 F8FF                          mov.b   #write,r0l
  293 program  C 0024 38B1                          mov.b   r0l,@mem_dir    ;set memory data port as output
  294 program  C 0026 F800                          mov.b   #0,r0l          ;clearing value
  295 program  C 0028 79050000                      mov.w   #0,r5           ;buffer pointer
  296 program  C 002C 38B3                          mov.b   r0l,@mem_data   ;set data
  297
  298 program  C 002E                       clear_buffer:
  299                                       ;clear buffer location
  300 program  C 002E 3DB6                          mov.b   r5l,@addr_lo
  301 program  C 0030 35BF                          mov.b   r5h,@addr_hi    ;set address
  302 program  C 0032 4B04                          bmi     wr_cs1
  303 program  C 0034                       wr_cs0:
  304 program  C 0034 F802                          mov.b   #wrcs0,r0l      ;write to chip 0
  305 program  C 0036 4002                          bra     wr_cont
  306 program  C 0038                       wr_cs1:
  307 program  C 0038 F801                          mov.b   #wrcs1,r0l      ;write to chip 1
  308 program  C 003A                       wr_cont:
  309 program  C 003A 38BA                          mov.b   r0l,@mem_ctrl   ;activate write pulse
  310 program  C 003C F807                          mov.b   #7,r0l
  311 program  C 003E 38BA                          mov.b   r0l,@mem_ctrl   ;de-activate write pulse
  312
  313                                       ;increment buffer pointer
  314 program  C 0040 8D01                          add.b   #1,r5l
  315 program  C 0042 44EA                          bcc     clear_buffer    ;loop until buffer cleared
  316 program  C 0044 8501                          add.b   #1,r5h
  317 program  C 0046 44E6                          bcc     clear_buffer    ;loop until buffer cleared
  318
```

Listing 2: START.LIS (continued)

```
  319                                   ;re-initialize buffer pointers
  320 program  C 0048 0D56                      mov.w   r5,r6           ;clear IDP and ODP
 *** H8/300 ASSEMBLER               VER 1.1 ***   03/20/91 08:11:20                                                  PAGE     2
PROGRAM NAME =                              Buffer Initialization Routine

  321
  322                                   ;initialize program constants and flags
  323
  324                                   ;initialize margin area
  325 program  C 004A F410                      mov.b   #h'10,r4h       ;margin = 16 bytes
  326
  327                                   ;initialize flags
  328 program  C 004C FC15                      mov.b   #h'15,r4l       ;set buf_mt, buf_init, & on_line active
  329
  330                                   ;initialize free-running timer for input watchdog timing
  331 program  C 004E 79000000                  mov.w   #0,r0
  332 program  C 0052 6B80FF92                  mov.w   r0,@frt_frc     ;reset counter
  333
  334 program  C 0056 F801                      mov.b   #1,r0l
  335 program  C 0058 3891                      mov.b   r0l,@frt_tcsr   ;clear counter on match a
  336
  337 program  C 005A F800                      mov.b   #0,r0l
  338 program  C 005C 3896                      mov.b   r0l,@frt_tcr    ;use phi/2
  339
  340 program  C 005E 790001F4                  mov.w   #500,r0
  341 program  C 0062 6B80FF94                  mov.w   r0,@frt_ocra    ;set count for 100 usec
  342
  343                                   ;initialize multi-function timer 0 for output strobe generation
  344 program  C 0066 F801                      mov.b   #1,r0l
  345 program  C 0068 38C8                      mov.b   r0l,@tmr0_tcr   ;use phi/8, no interrupts
  346
  347 program  C 006A F800                      mov.b   #0,r0l
  348 program  C 006C 38C9                      mov.b   r0l,@tmr0_tcsr  ;negative strobe
  349
  350 program  C 006E F800                      mov.b   #0,r0l
  351 program  C 0070 38CC                      mov.b   r0l,@tmr0_tcnt  ;clear counter
  352
  353 program  C 0072 F802                      mov.b   #2,r0l
  354 program  C 0074 38CB                      mov.b   r0l,@tmr0_tcorb
  355 program  C 0076 F805                      mov.b   #5,r0l
  356 program  C 0078 38CA                      mov.b   r0l,@tmr0_tcora ;generate strobe 2.4 usec wide
  357
  358                                   ;initialize multi-function timer 1 for initialization timing
  359 program  C 007A F801                      mov.b   #1,r0l
  360 program  C 007C 38D0                      mov.b   r0l,@tmr1_tcr   ;use phi/8, no interrupt
  361
  362 program  C 007E F806                      mov.b   #6,r0l
  363 program  C 0080 38D1                      mov.b   r0l,@tmr1_tcsr  ;negative strobe
  364
  365 program  C 0082 F800                      mov.b   #0,r0l
  366 program  C 0084 38D4                      mov.b   r0l,@tmr1_tcnt  ;set counter to 0
  367
  368 program  C 0086 F802                      mov.b   #2,r0l
  369 program  C 0088 38D3                      mov.b   r0l,@tmr1_tcorb
  370 program  C 008A F834                      mov.b   #h'34,r0l
  371 program  C 008C 38D2                      mov.b   r0l,@tmr1_tcora ;generate strobe 40 usec wide
  372
  373                                   ;initialize interrupt structure
  374 program  C 008E F807                      mov.b   #7,r0l          ;set maskable interrupts
  375 program  C 0090 38C6                      mov.b   r0l,@iscr       ; for falling edge
  376 program  C 0092 38C7                      mov.b   r0l,@ier        ;enable maskable interrupts
  377 program  C 0094 0700                      ldc     #h'00,ccr       ;global interrupt enable
  378
  379                                   ;enable init pulse timer
  380 program  C 0096 F800                      mov.b   #0,r0l
  381 program  C 0098 38D4                      mov.b   r0l,@tmr1_tcnt  ;reset init pulse count
  382
  383 program  C 009A F841                      mov.b   #h'41,r0l
  384 program  C 009C 38D0                      mov.b   r0l,@tmr1_tcr   ;use phi/8, interrupt on match a
  385
  386                                   ;test for initialization complete
  387 program  C 009E                   init_loop:
  388 program  C 009E 732C                      btst.b  #buf_init_flag,r4l      ;init done ?
  389 program  C 00A0 46FC                      bne     init_loop               ;no
  390
  391                                   ;enable input handshaking
  392 program  C 00A2 7FB27240                  bclr.b  #ibusy_bit,@in_hs       ;release input busy signal
  393
  394                                   ;set status indicators
  395 program  C 00A6 7FB27200                  bclr.b  #ready_bit,@stat_port   ;Light Ready LED
  396 program  C 00AA 7FB27210                  bclr.b  #online_bit,@stat_port  ;Light On Line LED
  397
```

Listing 2: START.LIS (continued)

```
  398                                   ;enable input delay timer
  399 program  C 00AE F808                      mov.b   #8,r0l
  400 program  C 00B0 3890                      mov.b   r0l,@frt_tier   ;enable delay timer interrupts
  401
*** H8/300 ASSEMBLER               VER 1.1 ***   03/20/91 08:11:20                                                  PAGE     3
PROGRAM NAME =                                Buffer Initialization Routine

  402                                   ;main loop
  403 program  C 00B2                   main:
  404 program  C 00B2 0180                      sleep                   ;wait for interrupts
  405
  406 program  C 00B4 40FC                      bra     main
  407
  408                                           .end
  *****TOTAL ERRORS       0
  *****TOTAL WARNINGS     0
```

Listing 3: INPUT.LIS

```
*** H8/300 ASSEMBLER               VER 1.1 ***   03/20/91 08:11:34                                    PAGE     1
PROGRAM NAME =                                   Date Input Routine

    1                                                .heading        “Date Input Routine”
    2
    3                                           ;H8/330 Print Buffer Routine
    4                                           ;version 2.0
    5
    6                                           ;written by:
    7                                           ;       Tom Hampton
    8                                           ;       Hitachi America, Ltd.
    9                                           ;       Application Engineering
   10
  248                                               .output         dbg,obj
  249                                               .print          nocref,nosct
  250
  251                                               .global         input_int
  252
  253 program  C 0000                               .section        program,code
  254
  255                                           ;Input /STB Interrupt Routine
  256
  257 program  C 0000                           input_int:
  258 program  C 0000 7FB27040                          bset.b  #ibusy_bit,@in_hs       ;set input busy signal
  259
  260                                           ;get input data
  261 program  C 0004                           get_data:
  262 program  C 0004 20BE                              mov.b   @in_port,r0h    ;read data
  263
  264                                           ;write data into memory buffer
  265 program  C 0006 F8FF                              mov.b   #write,r0l
  266 program  C 0008 38B1                              mov.b   r0l,@mem_dir    ;set port direction for write
  267 program  C 000A 30B3                              mov.b   r0h,@mem_data   ;set data
  268 program  C 000C 3DB6                              mov.b   r5l,@addr_lo
  269 program  C 000E 35BF                              mov.b   r5h,@addr_hi    ;output buffer address
  270 program  C 0010 4B04                              bmi     wr_cs1
  271 program  C 0012                           wr_cs0:
  272 program  C 0012 F802                              mov.b   #wrcs0,r0l      ;write to U3
  273 program  C 0014 4002                              bra     wr_cont
  274 program  C 0016                           wr_cs1:
  275 program  C 0016 F801                              mov.b   #wrcs1,r0l      ;write to U4
  276 program  C 0018                           wr_cont:
  277 program  C 0018 38BA                              mov.b   r0l,@mem_ctrl   ;activate write pulse
  278 program  C 001A F807                              mov.b   #7,r0l
  279 program  C 001C 38BA                              mov.b   r0l,@mem_ctrl   ;de-activate write pulse
  280 program  C 001E 0B05                              adds    #1,r5           ;increment input pointer
  281
  282                                           ;test for buffer full
  283 program  C 0020 0D53                              mov.w   r5,r3
  284 program  C 0022 1963                              sub.w   r6,r3           ;is IDP = ODP ?
  285 program  C 0024 4708                              beq     buff_full       ;yes
  286
  287                                           ;buffer is not full, but cannot be empty either
  288 program  C 0026 720C                              bclr.b  #buf_mt_flag,r4l        ;clear buffer empty flag
  289 program  C 0028 7FB27240                          bclr.b  #ibusy_bit,@in_hs       ;clear IBUSY signal
  290 program  C 002C 4006                              bra     clean_ret
  291
  292 program  C 002E                           buff_full:
  293 program  C 002E 701C                              bset.b  #buf_fl_flag,r4l        ;set buffer full flag
  294                                                                                   ; IBUSY remains set
  295 program  C 0030 7FB27220                          bclr.b  #buf_ful_bit,@stat_port ;turn Buffer Full LED ON
  296
  297                                           ;reset input delay timer
  298 program  C 0034                           clean_ret:
  299 program  C 0034 79000000                          mov.w   #0,r0
  300 program  C 0038 6B80FF92                          mov.w   r0,@frt_frc     ;reset delay timer
  301
  302                                           ;enable delay timer interrupts
  303 program  C 003C 7F917230                          bclr.b  #3,@frt_tcsr    ;clear compare flag
  304
  305 program  C 0040 5670                              rte
  306
  307                                               .end
  *****TOTAL ERRORS       0
  *****TOTAL WARNINGS     0
```

Listing 4: OUTPUT.LIS

```
*** H8/300 ASSEMBLER               VER 1.1 ***   03/20/91 08:11:43                                        PAGE     1
PROGRAM NAME =                                Data Output Routine

    1                                           .heading        “Data Output Routine”
    2
    3                                   ;H8/330 Print Buffer Routine
    4                                   ;version 2.0
    5
    6                                   ;written by:
    7                                   ;       Tom Hampton
    8                                   ;       Hitachi America, Ltd.
    9                                   ;       Application Engineering
   10
  248                                           .output         dbg,obj
  249                                           .print          nocref,nosct
  250
  251                                           .global         output_int
  252
  253 program  C 0000                           .section        program,code
  254
  255                                   ;Data Output Service
  256                                   ;input delay timer interrupt
  257
  258 program  C 0000                   output_int:
  259 program  C 0000 7FB27040                  bset.b  #ibusy_bit,@in_hs       ;set input busy signal
  260
  261                                   ;test for buffer empty
  262 program  C 0004 730C                      btst.b  #buf_mt_flag,r4l        ;is buffer empty ?
  263 program  C 0006 464E                      bne     ret1                    ;yes
  264
  265                                   ;test for output busy
  266 program  C 0008 7EB77300                  btst.b  #obusy_bit,@out_hs      ;output busy ?
  267 program  C 000C 4648                      bne     ret1                    ;yes
  268
  269                                   ;test for output hold
  270 program  C 000E 733C                      btst.b  #ohold_flag,r4l         ;output on hold ?
  271 program  C 0010 4644                      bne     ret1                    ;yes
  272
  273                                   ;get output buffer data
  274 program  C 0012 F800                      mov.b   #read,r0l
  275 program  C 0014 38B1                      mov.b   r0l,@mem_dir    ;set port direction for read
  276
  277 program  C 0016 3EB6                      mov.b   r6l,@addr_lo
  278 program  C 0018 36BF                      mov.b   r6h,@addr_hi    ;output buffer address
  279 program  C 001A 4B04                      bmi     rd_cs1
  280 program  C 001C                   rd_cs0:
  281 program  C 001C F806                      mov.b   #rdcs0,r0l      ;read from U3
  282 program  C 001E 4002                      bra     rd_cont
  283 program  C 0020                   rd_cs1:
  284 program  C 0020 F805                      mov.b   #rdcs1,r0l      ;read from U4
  285 program  C 0022                   rd_cont:
  286 program  C 0022 38BA                      mov.b   r0l,@mem_ctrl   ;activate chip select pulse
  287 program  C 0024 20B3                      mov.b   @mem_data,r0h   ;get output data
  288 program  C 0026 F807                      mov.b   #7,r0l
  289 program  C 0028 38BA                      mov.b   r0l,@mem_ctrl   ;de-activate write pulse
  290 program  C 002A 0B06                      adds    #1,r6   ;increment output pointer
  291
  292                                   ;output data to port
  293 program  C 002C 30BB                      mov.b   r0h,@out_port   ;output data
  294
  295                                   ;generate output data strobe
  296 program  C 002E F800                      mov.b   #0,r0l
  297 program  C 0030 38CC                      mov.b   r0l,@tmr0_tcnt  ;clear counter
  298 program  C 0032 28C9                      mov.b   @tmr0_tcsr,r0l  ;read flags
  299 program  C 0034 F816                      mov.b   #h’16,r0l
  300 program  C 0036 38C9                      mov.b   r0l,@tmr0_tcsr  ;generate a negative strobe
  301                                                                   ; 2.4 usec wide and clear flags
  302 program  C 0038 F841                      mov.b   #h’41,r0l
  303 program  C 003A 38C8                      mov.b   r0l,@tmr0_tcr   ;enable compare A interrupt
  304
  305                                   ;enable interrupts for strobe generation
  306 program  C 003C 0700                      ldc     #0,ccr          ;enable interrupts
  307
  308                                   ;wait for output strobe interrupt
  309 program  C 003E 0180                      sleep
  310
  311                                   ;disable interrupts
  312 program  C 0040 0780                      ldc     #h’80,ccr       ;mask interrupts
  313
  314                                   ;test for buffer empty
  315 program  C 0042 0D63                      mov.w   r6,r3           ;temporary work register
  316 program  C 0044 1953                      sub.w   r5,r3           ;is ODP = IDP ?
```

Listing 4: OUTPUT.LIS (continued)

```
  317 program  C 0046 470C                        beq     buff_mt ;yes
  318
*** H8/300 ASSEMBLER               VER 1.1 ***   03/20/91 08:11:43                                                PAGE     2
PROGRAM NAME =                                  Data Output Routine

  319                                   ;test for in full area
  320 program  C 0048 084B                        add.b   r4h,r3l         ;is buffer still full ?
  321 program  C 004A 4516                        bcs     ret2            ;yes
  322
  323 program  C 004C 721C                        bclr.b  #buf_fl_flag,r4l        ;clear buffer full flag
  324 program  C 004E 7FB27020                    bset.b  #buf_ful_bit,@stat_port ;turn Buffer Full LED OFF
  325 program  C 0052 4002                        bra     ret1
  326
  327                                   ;set buffer empty flag
  328 program  C 0054                   buff_mt:
  329 program  C 0054 700C                        bset.b  #buf_mt_flag,r4l        ;set buffer empty flag
  330
  331                                   ;should IBUSY be cleared ?
  332 program  C 0056                   ret1:
  333 program  C 0056 734C                        btst.b  #online_flag,r4l        ;is buffer online ?
  334 program  C 0058 4708                        beq     ret2                    ;no
  335
  336 program  C 005A 731C                        btst.b  #buf_fl_flag,r4l        ;is buffer full ?
  337 program  C 005C 4604                        bne     ret2                    ;yes
  338
  339                                   ;clear IBUSY signal
  340 program  C 005E 7FB27240                    bclr.b  #ibusy_bit,@in_hs       ;set IBUSY inactive
  341
  342 program  C 0062                   ret2:
  343 program  C 0062 79000000                    mov.w   #0,r0
  344 program  C 0066 6B80FF92                    mov.w   r0,@frt_frc     ;reset delay timer
  345
  346                                   ;enable delay timer interrupts
  347 program  C 006A 7F917230                    bclr.b  #3,@frt_tcsr    ;clear compare flag
  348
  349 program  C 006E 5670                        rte
  350
  351                                           .end
  *****TOTAL ERRORS       0
  *****TOTAL WARNINGS     0
```

Listing 5: OUT-STB.LIS

```
*** H8/300 ASSEMBLER               VER 1.1 ***   03/20/91 08:11:54                                        PAGE     1
PROGRAM NAME =                                Output STB Routine

    1                                             .heading        “Output STB Routine”
    2
    3                                       ;H8/330 Print Buffer Routine
    4                                       ;version 2.0
    5
    6                                       ;written by:
    7                                       ;       Tom Hampton
    8                                       ;       Hitachi America, Ltd.
    9                                       ;       Application Engineering
   10
  248                                             .output         dbg,obj
  249                                             .print          nocref,nosct
  250
  251                                             .global         ostb_int
  252
  253 program  C 0000                             .section        program,code
  254
  255                                       ;Output Strobe interrupt
  256 program  C 0000                       ostb_int:
  257 program  C 0000 F801                        mov.b   #1,r0l
  258 program  C 0002 38C8                        mov.b   r0l,@tmr0_tcr   ;disable further timer interrupts
  259
  260 program  C 0004 28C9                        mov.b   @tmr0_tcsr,r0l  ;read flags
  261 program  C 0006 F81A                        mov.b   #h’1a,r0l
  262 program  C 0008 38C9                        mov.b   r0l,@tmr0_tcsr  ;clear flags, outputs to high level
  263
  264 program  C 000A 5670                        rte
  265
  266                                             .end
  *****TOTAL ERRORS       0
  *****TOTAL WARNINGS     0
```

Listing 6: IN-INIT.LIS

```
*** H8/300 ASSEMBLER               VER 1.1 ***   03/20/91 08:12:01                                        PAGE     1
PROGRAM NAME =                              Input INIT Pulse Service Routine

    1                                             .heading        “Input INIT Pulse Service Routine”
    2
    3                                       ;H8/330 Print Buffer Routine
    4                                       ;version 2.0
    5
    6                                       ;written by:
    7                                       ;       Tom Hampton
    8                                       ;       Hitachi America, Ltd.
    9                                       ;       Application Engineering
   10
  248                                             .output         dbg,obj
  249                                             .print          nocref,nosct
  250
  251                                             .global         iinit_int,start
  252
  253 program  C 0000                             .section        program,code
  254
  255                                       ;disable any timer interrupts
  256 program  C 0000                       iinit_int:
  257 program  C 0000 F801                        mov.b   #1,r0l
  258 program  C 0002 38C8                        mov.b   r0l,@tmr0_tcr   ;disable output strobe interrupts
  259 program  C 0004 38D0                        mov.b   r0l,@tmr1_tcr   ;disable init pulse interrupts
  260 program  C 0006 3890                        mov.b   r0l,@frt_tier   ;disable input delay interrupts
  261
  262                                       ;jump to beginning
  263 program  C 0008 5A000000                    jmp     @start          ;jump to initialization routine
  264
  265                                             .end
  *****TOTAL ERRORS       0
  *****TOTAL WARNINGS     0
```

Listing 7: OUT-INIT.LIS

```
*** H8/300 ASSEMBLER               VER 1.1 ***   03/20/91 08:12:08                                        PAGE     1
PROGRAM NAME =                                  INIT Pulse Output Routine

    1                                                   .heading        “INIT Pulse Output Routine”
    2
    3                                           ;H8/330 Print Buffer Routine
    4                                           ;version 2.0
    5
    6                                           ;written by:
    7                                           ;       Tom Hampton
    8                                           ;       Hitachi America, Ltd.
    9                                           ;       Application Engineering
   10
  248                                                   .output         dbg,obj
  249                                                   .print          nocref,nosct
  250
  251                                                   .global         oinit_int
  252
  253 program  C 0000                                   .section        program,code
  254
  255                                           ;output INIT signal interrupt
  256 program  C 0000                           oinit_int:
  257                                           ;disable further timer interrupts
  258 program  C 0000 F901                              mov.b   #1,r1l
  259 program  C 0002 39D0                              mov.b   r1l,@tmr1_tcr   ;use phi/8, no interrupts
  260
  261                                           ;clear match flag
  262 program  C 0004 7FD17260                          bclr.b  #6,@tmr1_tcsr   ;clear compare match a flag
  263
  264                                           ;clear OINIT\ signal
  265 program  C 0008 F900                              mov.b   #0,r1l
  266 program  C 000A 39D1                              mov.b   r1l,@tmr1_tcsr  ;no more strobes
  267
  268                                           ;clear oinit flag
  269 program  C 000C 722C                              bclr.b  #buf_init_flag,r4l      ;clear oinit flag
  270
  271 program  C 000E 5670                              rte
  272
  273                                                   .end
  *****TOTAL ERRORS       0
  *****TOTAL WARNINGS     0
```

Listing 8: ONLINE.LIS

```
*** H8/300 ASSEMBLER               VER 1.1 ***   03/20/91 08:12:16                                                    PAGE     1
PROGRAM NAME =                                Online Pushbutton Service Routine

    1                                           .heading        “Online Pushbutton Service Routine”
    2
    3                                   ;H8/330 Print Buffer Routine
    4                                   ;version 2.0
    5
    6                                   ;written by:
    7                                   ;       Tom Hampton
    8                                   ;       Hitachi America, Ltd.
    9                                   ;       Application Engineering
   10
  248                                           .output         dbg,obj
  249                                           .print          nocref,nosct
  250
  251                                           .global         online_int
  252
  253 program  C 0000                           .section        program,code
  254
  255                                   ;on line pushbutton test
  256 program  C 0000                   online_int:
  257
  258                                   ;set input port busy
  259 program  C 0000 7FB27040                  bset.b  #ibusy_bit,@in_hs       ;set IBUSY active
  260
  261                                   ;test online switch
  262 program  C 0004                   test_sw:
  263 program  C 0004 7EC17320                  btst.b  #online_sw_bit,@in_hs2  ;test online switch
  264 program  C 0008 47FA                      beq     test_sw                 ;still low
  265                                                                           ; will not go further
  266                                                                           ; until released
  267
  268                                   ;test online status
  269 program  C 000A 734C                      btst.b  #online_flag,r4l        ;test online status
  270 program  C 000C 4708                      beq     put_online              ;currently offline
  271
  272                                   ;currently online
  273 program  C 000E                   put_offline:
  274 program  C 000E 7FB27010                  bset.b  #online_bit,@stat_port  ;clear Online LED
  275 program  C 0012 724C                      bclr.b  #online_flag,r4l        ;clear online status
  276
  277 program  C 0014                   just_ret:
  278 program  C 0014 5670                      rte
  279
  280                                   ;currently offline
  281 program  C 0016                   put_online:
  282 program  C 0016 7FB27210                  bclr.b  #online_bit,@stat_port  ;set Online LED
  283 program  C 001A 704C                      bset.b  #online_flag,r4l        ;set online status
  284
  285                                   ;should IBUSY be cleared ?
  286 program  C 001C 731C                      btst.b  #buf_fl_flag,r4l        ;is buffer full
  287 program  C 001E 46F4                      bne     just_ret                ;yes, IBUSY should remain active
  288
  289                                   ;clear input port busy
  290 program  C 0020 7FB27240                  bclr.b  #ibusy_bit,@in_hs       ;set IBUSY inactive
  291
  292 program  C 0024 5670                      rte
  293
  294                                           .end
 *****TOTAL ERRORS       0
 *****TOTAL WARNINGS     0
```

Listing 9: PAUSE.LIS

```
*** H8/300 ASSEMBLER               VER 1.1 ***   03/20/91 08:12:24                                        PAGE     1
PROGRAM NAME =                                  Pause Pushbutton Service Routine

    1                                           .heading        “Pause Pushbutton Service Routine”
    2
    3                                   ;H8/330 Print Buffer Routine
    4                                   ;version 2.0
    5
    6                                   ;written by:
    7                                   ;       Tom Hampton
    8                                   ;       Hitachi America, Ltd.
    9                                   ;       Application Engineering
   10
  248                                           .output         dbg,obj
  249                                           .print          nocref,nosct
  250
  251                                           .global         pause_int
  252
  253 program  C 0000                           .section        program,code
  254
  255                                   ;pause pushbutton test
  256 program  C 0000                   pause_int:
  257
  258                                   ;set input port busy
  259 program  C 0000 7FB27040                  bset.b  #ibusy_bit,@in_hs       ;set IBUSY active
  260
  261                                   ;test online switch
  262 program  C 0004                   test_sw:
  263 program  C 0004 7EC17310                  btst.b  #pause_sw_bit,@in_hs2   ;test pause switch
  264 program  C 0008 47FA                      beq     test_sw                 ;still low
  265                                                                           ; will not go further
  266                                                                           ; until released
  267
  268                                   ;test hold status
  269 program  C 000A 733C                      btst.b  #ohold_flag,r4l ;test hold status
  270 program  C 000C 470C                      beq     put_on_hold     ;currently not on hold
  271
  272                                   ;currently on hold
  273 program  C 000E                   put_off_hold:
  274 program  C 000E 7FB27030                  bset.b  #ohold_bit,@stat_port   ;clear Output Hold LED
  275 program  C 0012 723C                      bclr.b  #ohold_flag,r4l ;clear hold status
  276
  277                                   ;enable delay timer interrupts
  278 program  C 0014 F808                      mov.b   #8,r0l
  279 program  C 0016 3890                      mov.b   r0l,@frt_tier   ;enable delay timer interrupts
  280
  281 program  C 0018 400E                      bra     pause_cont
  282
  283                                   ;currently off hold
  284 program  C 001A                   put_on_hold:
  285 program  C 001A 7FB27230                  bclr.b  #ohold_bit,@stat_port   ;set Output Hold LED
  286 program  C 001E 703C                      bset.b  #ohold_flag,r4l         ;set hold status
  287
  288                                   ;disable delay timer interrupts
  289 program  C 0020 F800                      mov.b   #0,r0l
  290 program  C 0022 3890                      mov.b   r0l,@frt_tier   ;disable delay timer interrupts
  291 program  C 0024 7F917230                  bclr.b  #3,@frt_tcsr    ;clear compare flag in case
  292
  293 program  C 0028                   pause_cont:
  294                                   ;should IBUSY be cleared ?
  295                                   ;test for online first
  296 program  C 0028 734C                      btst.b  #online_flag,r4l        ;is buffer offline ?
  297 program  C 002A 4708                      beq     pause_ret               ;yes, keep IBUSY set
  298
  299                                   ;test for buffer full
  300 program  C 002C 731C                      btst.b  #buf_fl_flag,r4l        ;is buffer full ?
  301 program  C 002E 4604                      bne     pause_ret               ;yes, keep IBUSY set
  302
  303                                   ;clear IBUSY
  304 program  C 0030 7FB27240                  bclr.b  #ibusy_bit,@in_hs       ;set IBUSY inactive
  305
  306 program  C 0034                   pause_ret:
  307 program  C 0034 5670                      rte
  308
  309                                           .end
  *****TOTAL ERRORS       0
  *****TOTAL WARNINGS     0
```

Listing 10: BUFFER.INC

```
;H8/330 Printer Buffer Program
;revision 2.0

;Register Usage
;       R0H = Buffer Data
;       R4L = Status Flags
;       R4H = Buffer Margin (16 bytes)
;       R5  = Input Buffer Pointer
;       R6  = Output Buffer Pointer

;buffer control flags (R4L)
online_flag     .equ    4               ;       -4- On Line Flag
ohold_flag      .equ    3               ;       -3- Output Hold Flag
buf_init_flag   .equ    2               ;       -2- Buffer Init Flag
buf_fl_flag     .equ    1               ;       -1- Buffer Full Flag
buf_mt_flag     .equ    0               ;       -0- Buffer Empty Flag

;Port 1 Usage
in_hs           .equ    p1_dr           ;Input Port Control (output)
stat_port       .equ    p1_dr           ;Status Indicators
ibusy_bit       .equ    4               ;       -4- IBUSY (output)
ohold_bit       .equ    3               ;       -3- Output Hold LED (output)
buf_ful_bit     .equ    2               ;       -2- Buffer Full LED (output)
online_bit      .equ    1               ;       -1- On Line LED (output)
ready_bit       .equ    0               ;       -0- Ready LED (output)

;Port 2 Usage
mem_data        .equ    p2_dr           ;Data (input/output)
mem_dir         .equ    p2_ddr
write           .equ    h'ff            ;write direction
read            .equ    0               ;read direction

;Port 3 Usage
addr_lo         .equ    p3_dr           ;Address, Low Byte (output)

;Port 4 Usage
out_hs          .equ    p4_dr           ;Output Port Handshake
oinit_bit       .equ    4               ;       -4- OINIT\ (output)
                                        ;       -1- OSTB\ (TMO0)
obusy_bit       .equ    0               ;       -0- OBUSY (input)

;Port 5 Usage
mem_ctrl        .equ    p5_dr           ;Memory Buffer Control
                                        ;       -2- WE\ (output)
                                        ;       -1- CS1\ (output)
                                        ;       -0- CS0\ (output)
wrcs1           .equ    1               ;write to chip 1
wrcs0           .equ    2               ;write to chip 0
rdcs1           .equ    5               ;read from chip 1
rdcs0           .equ    6               ;read from chip 0

;Port 6 Usage
out_port        .equ    p6_dr           ;Output Port

;Port 7 Usage
in_port         .equ    p7_dr           ;Input Port

;Port 8 Usage
addr_hi         .equ    p8_dr           ;Address, High Byte (output)

;Port 9 Usage
in_hs2          .equ    p9_dr           ;Port Control (inputs)
online_sw_bit   .equ    2               ;       -2- ONLINE (input, IRQ0)
pause_sw_bit    .equ    1               ;       -1- HOLD\ (input, IRQ1)
istb_bit        .equ    0               ;       -0- ISTB\ (input, IRQ2)

;maskable interrupts
online          .equ    0               ;irq0
pause           .equ    1               ;irq1
istb            .equ    2               ;irq2
```

Listing 11: H8330.INC

```
;H8/330 Port Definitions

;I/O Port Address
p1_ddr        .equ          h'ffb0
p2_ddr        .equ          h'ffb1
p3_ddr        .equ          h'ffb4
p4_ddr        .equ          h'ffb5
p5_ddr        .equ          h'ffb8
p6_ddr        .equ          h'ffb9
p8_ddr        .equ          h'ffbd
p9_ddr        .equ          h'ffc0

p1_dr         .equ          h'ffb2
p2_dr         .equ          h'ffb3
p3_dr         .equ          h'ffb6
p4_dr         .equ          h'ffb7
p5_dr         .equ          h'ffba
p6_dr         .equ          h'ffbb
p7_dr         .equ          h'ffbe
p8_dr         .equ          h'ffbf
p9_dr         .equ          h'ffc1

;System Control Registers
syscr         .equ          h'ffc4
mdcr          .equ          h'ffc5
iscr          .equ          h'ffc6
ier           .equ          h'ffc7

;Free-Running Timer
frt           .equ          h'ff90
frt_tier      .equ          h'ff90
frt_tcsr      .equ          h'ff91
frt_frc       .equ          h'ff92
frt_frch      .equ          h'ff92
frt_frcl      .equ          h'ff93
frt_ocra      .equ          h'ff94
frt_ocrah     .equ          h'ff94
frt_ocral     .equ          h'ff95
frt_ocrb      .equ          h'ff94
frt_ocrbh     .equ          h'ff94
frt_ocrbl     .equ          h'ff95
frt_tcr       .equ          h'ff96
frt_tocr      .equ          h'ff97
frt_icra      .equ          h'ff98
frt_icrah     .equ          h'ff98
frt_icral     .equ          h'ff99
frt_icrb      .equ          h'ff9a
frt_icrbh     .equ          h'ff9a
frt_icrbl     .equ          h'ff9b
frt_icrc      .equ          h'ff9c
frt_icrch     .equ          h'ff9c
frt_icrcl     .equ          h'ff9d
frt_icrd      .equ          h'ff9e
frt_icrdh     .equ          h'ff9e
frt_icrdl     .equ          h'ff9f

;Pulse-Width Modulation Timers
pwm0          .equ          h'ffa0
pwm0_tcr      .equ          h'ffa0
pwm0_dtr      .equ          h'ffa1
pwm0_tcnt     .equ          h'ffa2
pwm1          .equ          h'ffa4
pwm1_tcr      .equ          h'ffa4
pwm1_dtr      .equ          h'ffa5
pwm1_tcnt     .equ          h'ffa6

;Multi-Function Timers
tmr0          .equ          h'ffc8
tmr0_tcr      .equ          h'ffc8
tmr0_tcsr     .equ          h'ffc9
tmr0_tcora    .equ          h'ffca
tmr0_tcorb    .equ          h'ffcb
tmr0_tcnt     .equ          h'ffcc
tmr1          .equ          h'ffd0
tmr1_tcr      .equ          h'ffd0
tmr1_tcsr     .equ          h'ffd1
tmr1_tcora    .equ          h'ffd2
tmr1_tcorb    .equ          h'ffd3
tmr1_tcnt     .equ          h'ffd4

;Serial Communications Interface
sci           .equ          h'ffd8
sci_smr       .equ          h'ffd8
sci_brr       .equ          h'ffd9
```

Listing 11: H8330.INC (continued)

```
sci_scr      .equ          h'ffda
sci_tdr      .equ          h'ffdb
sci_ssr      .equ          h'ffdc
sci_rdr      .equ          h'ffdd

;A/D Converter
adc          .equ          h'ffe0
adc_a        .equ          h'ffe0
adc_b        .equ          h'ffe2
adc_c        .equ          h'ffe4
adc_d        .equ          h'ffe6
adc_adcsr    .equ          h'ffe8
adc_adcr     .equ          h'ffea

;Dual-Port RAM
dpram        .equ          h'fff0
pccsr        .equ          h'fff0
pcdr0        .equ          h'fff1
pcdr2        .equ          h'fff2
pcdr3        .equ          h'fff3
pcdr4        .equ          h'fff4
pcdr5        .equ          h'fff5
pcdr6        .equ          h'fff6
pcdr7        .equ          h'fff7
pcdr8        .equ          h'fff8
pcdr9        .equ          h'fff9
pcdr10       .equ          h'fffa
pcdr11       .equ          h'fffb
pcdr12       .equ          h'fffc
pcdr13       .equ          h'fffd
pcdr14       .equ          h'fffe
pcdr15       .equ          h'ffff

;Memory Definitions
code_start   .equ          h'0040
end_rom      .equ          h'3fff
ram_start    .equ          h'fd80
end_ram      .equ          h'ff7f
top_ram      .equ          h'ff80

;Interrupt Vector Locations
nmi_vec      .equ          h'0006                  ;nmi
irq0_vec     .equ          h'0008                  ;irq0
irq1_vec     .equ          h'000a                  ;irq1
irq2_vec     .equ          h'000c                  ;irq2
irq3_vec     .equ          h'000e                  ;irq3
irq4_vec     .equ          h'0010                  ;irq4
irq5_vec     .equ          h'0012                  ;irq5
irq6_vec     .equ          h'0014                  ;irq6
irq7_vec     .equ          h'0016                  ;irq7
icia_vec     .equ          h'0018                  ;frt input capture a
icib_vec     .equ          h'001a                  ;frt input capture b
icic_vec     .equ          h'001c                  ;frt input capture c
icid_vec     .equ          h'001e                  ;frt input capture d
ocia_vec     .equ          h'0020                  ;frt output compare a
ocib_vec     .equ          h'0022                  ;frt output compare b
fovi_vec     .equ          h'0024                  ;frt overflow
cmi0a_vec    .equ          h'0026                  ;mft0 output compare a
cmi0b_vec    .equ          h'0028                  ;mft0 output compare b
ovi0_vec     .equ          h'002a                  ;mft0 overflow
cmi1a_vec    .equ          h'002c                  ;mft1 output compare a
cmi1b_vec    .equ          h'002e                  ;mft1 output compare b
ovi1_vec     .equ          h'0030                  ;mft1 overflow
mrei_vec     .equ          h'0032                  ;dpram read end
mwei_vec     .equ          h'0034                  ;dpram write end
eri_vec      .equ          h'0036                  ;receive error
rxi_vec      .equ          h'0038                  ;receive data available
txi_vec      .equ          h'003a                  ;transmit buffer empty
adi_vec      .equ          h'003c                  ;a/d complete
```