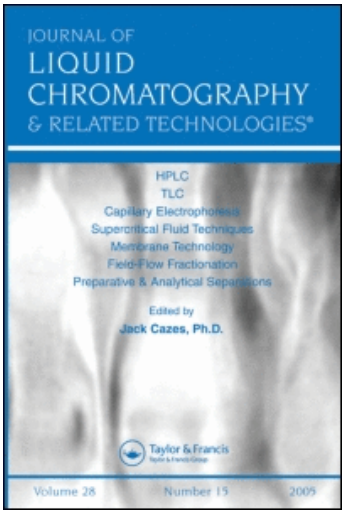

Journal of Liquid Chromatography & Related Technologies
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713597273

Hardware for Microprocessor Controlled HPLC: Interfacing of an Interval Timer and Interrupt Controller to the "S100 Bus System"

David B. Smoll[a]; Ram P. Singhal[a]
[a] Department of Chemistry, Wichita State University, Wichita, Kansas

To cite this Article Smoll, David B. and Singhal, Ram P.(1983) 'Hardware for Microprocessor Controlled HPLC: Interfacing of an Interval Timer and Interrupt Controller to the "S100 Bus System"', Journal of Liquid Chromatography & Related Technologies, 6: 11, 2095 — 2108
To link to this Article: DOI: 10.1080/01483918308066563
URL: http://dx.doi.org/10.1080/01483918308066563

JOURNAL OF LIQUID CHROMATOGRAPHY, 6(11), 2095-2108 (1983)

HARDWARE FOR MICROPROCESSOR CONTROLLED HPLC: INTERFACING OF AN INTERVAL TIMER AND INTERRUPT CONTROLLER TO THE "S100 BUS SYSTEM"

David B. Smoll and Ram P. Singhal*
Department of Chemistry, Wichita State University,
Wichita, Kansas 67208

ABSTRACT. A hardware interface for the 8253 Interval Timer and the 8259 Interrupt Controller to the standard "S100 Interface Bus System" is described herein. The interface allows the three timers of the 8253 Interval Timer to be synchronized by either an external trigger or an internal trigger issued by the microprocessor. This permits synchronized control of external devices either by the microprocessor, manually, or otherwise externally. Furthermore, two of the three timers are driven by two independent programmable clocks. This enhances their range, in contrast to a fixed clock, while maintaining the lowest possible error. The addition of the 8259 Interrupt Controller further supplements the flexibility of this interface. It permits the synchronization of software along with the external devices and permits the timers to be used in a "stopwatch" fashion by allowing them and the external trigger to initiate an "interrupt".

INTRODUCTION

The advent of "single board" microprocessors such as the "Sym", "Kim" and "Explorer 85" have brought automated HPLC systems within the economic reach of many laborators. These inexpensive programmable controllers are easily coupled to a larger computer in

*Address all correspondence to RPS.

 0148-3919/83/0611-2095$3.50/0

a master/slave configuration. The microprocessor, as the slave, is used to control the chromatography apparatus while collecting data for transfer to a master computer for subsequent processing and print-out.

A computer-microprocessor network of this style requires two interfaces, one between the microprocessor and the apparatus to control the system; the other between the microprocessor and the computer to perform data transfer. Fortunately, the major components for various versions of the latter interface are included on most computers and microprocessors. These are usually accessed directly through the "user" or "Peripheral Port". If not, a "Peripheral Port Interface" chip provides a simple, straight-forward interface.

The other interface, between the microprocessor and the chromatography apparatus, is rarely provided and is much more involved. It consists of any number of analog-digital and digital-analog converters, device select decoders and drivers, data registers, timers and controllers to name only a few. However, the literature on such interfaces to a large extent is limited to manufacturer's data books on the individual integrated circuits. The hardware and actual circuitry associated with these chips are somewhat neglected. To fill this gap, it is the purpose of this paper to describe a software programmable interval timer-interrupt controller interface.

The need for such an interface is self-evident. In addition to providing the three independent timers of the 8253 Interval

Timer (I.T.) the interface also provides a means by which an external device (the HPLC apparatus) can communicate with the microprocessor's central processing unit (CPU) by including the 8259 Interrupt Controller (I.C.). Through this controller, devices such as thermostats for constant temperature baths, analog-digital converters for monitoring the output of absorbance monitors and pH meters, pressure sensors in pumping systems, and even fluid level sensors in eluant reservoirs can signal the CPU. The 8259 I.C. is not an "input port" for digital data but rather an interrupt signal source. After receiving an interrupt request signal from a device, the controller issues an interrupt signal to the CPU followed by a subroutine address to which the program jumps. The address and the interrupt priorities are both software programmable.

This, however, is not the only function of the interface. The main intent is to provide the microprocessor with several programmable timers and necessary control and synchronization circuitry for accurately timed and synchronized intervals. Such timed intervals are a must in microprocessor controlled HPLC systems if the potential speed and accuracy of such a system is to be achieved. This is particularly true when making periodic measurements. Even a small error in timing becomes substantial when several thousand measurements are taken. However, this problem is eliminated by this interface. As a consequence, the timers may be programmed in a "free-running" mode without losing the initial accuracy and synchronization. This simplifies

the programming when taking periodic readings of absorbance for integration of peak areas. Furthermore, a provision is provided in the timer control circuitry to enable a timer to be programmed as a "stopwatch", thus allowing the computation of flow rates by the microprocessor and consequently the subsequent computation of retention volumes and times, as well as fraction size.

The remainder of the interface consists of three registers and associated circuitry which are decoded as two output ports and one input port as shown in Diagrams 1a and 1b. The two output ports (timer control register and clock select register) provide software control of all three gates and two of the three independent clocks of the 8253 I.T. The Timer Control Register also provides control of two "External Timed Outputs" and the mode of synchronization (internal or external) of the timers, external devices and software. The third register (Timer Status Register) provides the CPU with the status of the gates and outputs of the timers and the status of the external trigger.

Furthermore, this interface takes full advantage of the microprocessor's reaction time and speed by allowing machine language level programming. This eliminates much of the redundancy in the machine language program generated when using a higher level language such as "FORTRAN" or "BASIC".

MATERIALS

The 8253 Interval Timer and 8259 Interrupt Controller (Intel Corp., Sanat Clara, CA) were purchased from the Digi-Key Corp.,

Thief River Falls, MN. All other integrated circuits used in this interface are of the SN74LS' Series (Texas Instruments, Inc., Dallas, TX). The microprocessor used is the "Explorer 85" single board microprocessor, ASK II terminal version with levels A, B, S, D, and E (Newtronics Research and Development LTD, New Milford, CT). It uses the 8085 CPU and is equipped with a standard "S100 Bus System" and interface card cage. The interface cards (Universal Microcomputer Processor Plugboard, No. 8800V, Vector Electronic Co., Inc.) were obtained from Jameco Electronics, Belmont, CA and are designed for "wire-wrap" assembly. The "Explorer 85" is interfaced to a ASR35 teletype (Teletype Corp., Skokie, IL) with a 20 milliamp loop. All programming is done on the machine language level of the 8085 CPU.

DISCUSSION

Upon initialization of the Explorer 85 Microprocessor, pressing the "Reset Key", a "Reset Signal" is sent out on the Power On and Clear ($\overline{\text{POC}}$) line of the "S100 Bus". This in turn causes two events to occur in the 8253 I.T. interface hardware. All output registers are cleared and all flip-flops are reset. This causes all gates to the timers of the 8253 I.T. to go low, which turns off each timer (1); the two programmable clocks are set to 1.0 megahertz and all "external" circuits are disabled. (A low condition of a line or signal is defined as a voltage between 0.45 and 0.0 volts on that line. It is synonymous with a digital logic 0 or false condition.) Consequently, the external trig-

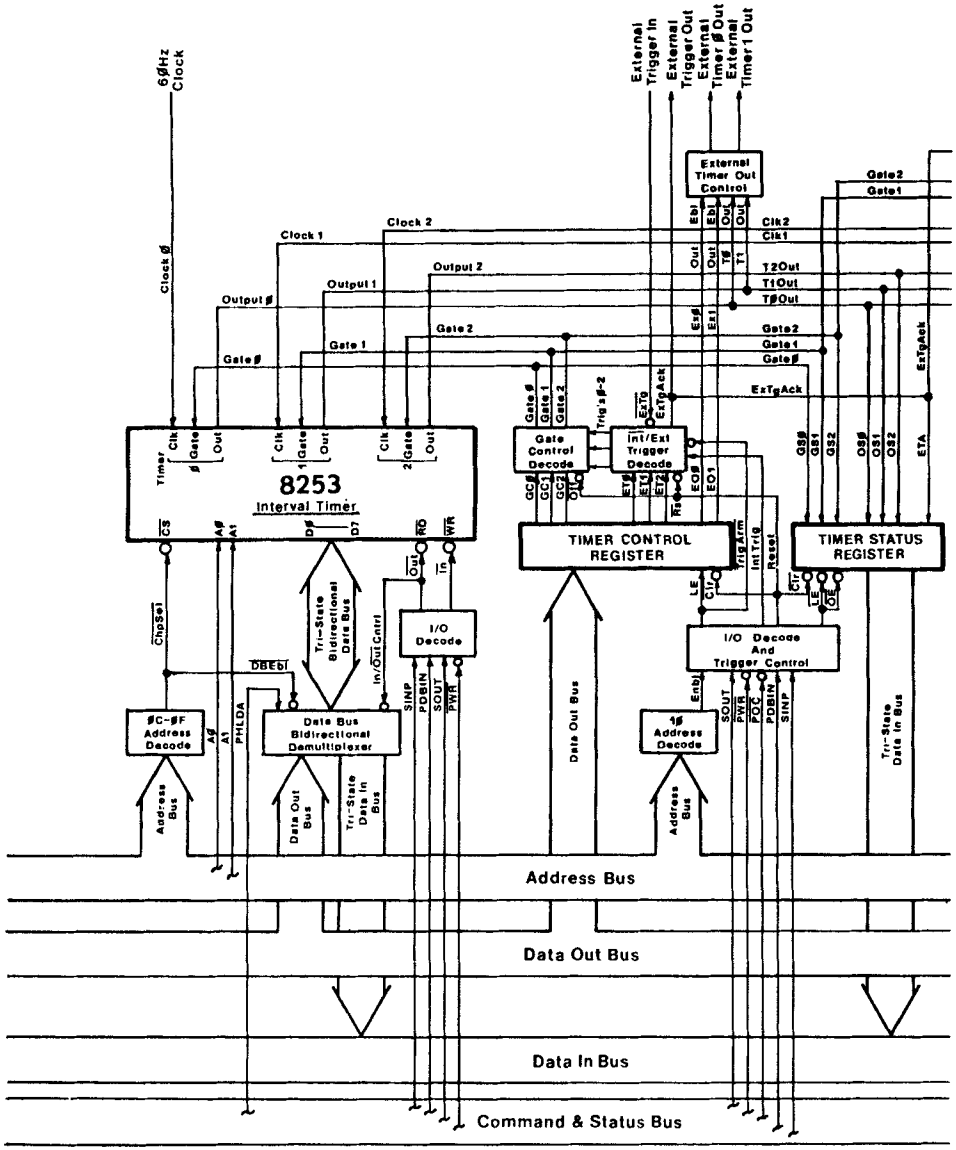

DIAGRAM 1

Microprocessor Controlled HPLC. A schematic view of the flow diagram showing the interfacing of the interval timer, 8253 (Diagram 1a) and the interrupt controller, 8259 (Diagram 1b) to the "S100 Bus System". See the text for details. (Write to the author RPS for a clear diagram and details.)

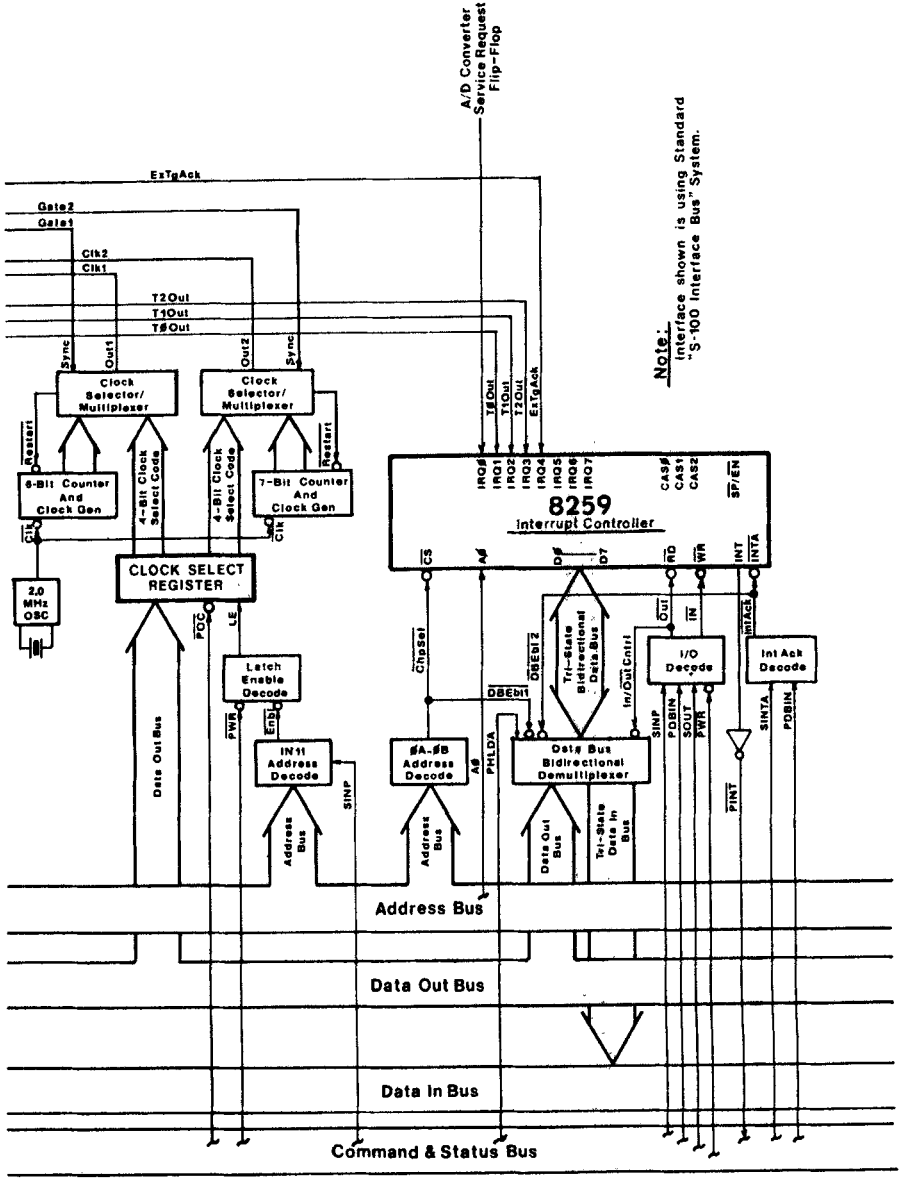

DIAGRAM I (continued)

ger is cleared and disabled and the External-Timer-Outputs are reset low; thus preventing the timers or the external circuitry from causing a system interrupt through the 8259 I.C. or triggering an external device to which they are connected.

The 8253 I.T. and associated registers are now ready to be programmed. The 8253 I.T. is programmed in the usual way (1). Its Mode Control Register and the three counters are "loaded" with "output" instructions. However, the timers do not start since all gates are maintained low.

1. The Timer Control Register.

a. *Control of the gates of the timers.* The gates are controlled by the Timer Control Register as illustrated in Diagram 1a. The outputs of the Timer Control Register are divided into three sets as shown in Figure 1. The first three bits of the register (D_0 to D_2) control the status of the gates to the timers located in the 8253 I.T. After initialization, each bit (D_0 to D_2) is reset low. As a consequence, this inhibits the counters in the timers, in effect, keeping them turned off. To start a particular timer the bit corresponding to that gate is set high. (A high condition of a line or signal is defined as a voltage between 2.4 and 5.0 volts on that line. It is synonymous with a digital logic 1 or true condition.) A specific timer(s) can be turned on or off by setting or resetting the appropriate bit(s) since the state of the bits is independent. In this way the computer can control the running of the timers directly in a synchronous manner.

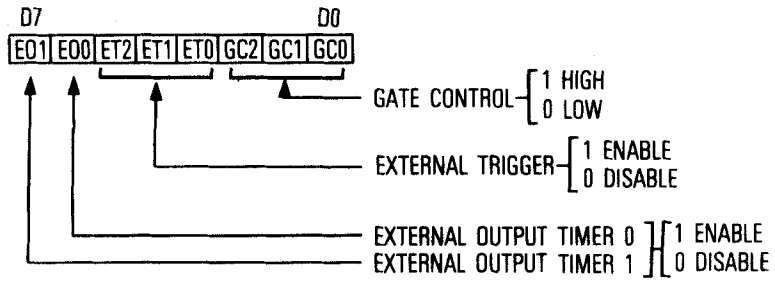

FIGURE 1

The Time Control Register (see Diagram 1a for its relation with other components). The gates and synchronization of the timers are controlled by the register.

b. *Synchronization of the timers.* The synchronization of the timers is achieved by generating a trigger pulse which clocks the data in bits D_0, D_1 and D_2 to the outputs of three "D" filp-flops which are connected to the gates of the timers (see Diagram 1a). The trigger pulse can be generated either internally or externally. This trigger pulse generating source is controlled by the next three bits of the Timer Control Register (D_3 to D_5) (see Figure 1). In the case where the bit (D_3 to D_5) is low, the trigger pulse for the corresponding gate is generated internally at the end of the "output" instruction cycle which loads the Timer Control Register. However, when the bit for the desired gate is set high, the internally generated trigger is inhibited and the external trigger is enabled for that gate. The external trigger can be generated at any time at the operator's discretion by grounding the "External Trigger In" line after setting one or more of the trigger control bits. In addition, after generating

the external trigger, an External Trigger Acknowledge signal is generated which can be used to initiate an interrupt through the 8259 I.C. and thus synchronize software with the External Trigger. Furthermore, this acknowledge signal can be used to control an external device via the "External Trigger Out" line.

c. *Control of external devices by the timers.* The last two bits of this register (D_6 and D_7) are an External Timer Output Enable (see Diagram 1 and Figure 1). The setting of bit D_6 or D_7 (or both) make outputs of Timer 0 and Timer 1, respectively, available to an external device. Consequently, these two timers can be used to control and synchronize an external devide or event such as the timed collection of fractions or the switching of eluant after a prescribed length of time.

2. The Clock Select Register.

a. *Selection of clock frequency.* The second output port is the Clock Select Register (see Diagram 1b). This register is divided into two 4-bit registers as shown in Figure 2. The first four bits (D_0 to D_3) determine the clock frequency for Timer 1 while the second four bits (D_4 to D_9) determine the clock frequency for Timer 2. (Timer 0 has a fixed clock frequency of 60 hertz derived from the AC supply of the power supply.) The actual frequency is selected by the last three bits of each 4-bit register. The clock periods and associated three bit code for each timer are shown in Figure 3. The periods are prime numbers listed in micro-seconds. These prime numbers periods permit a wide variety of

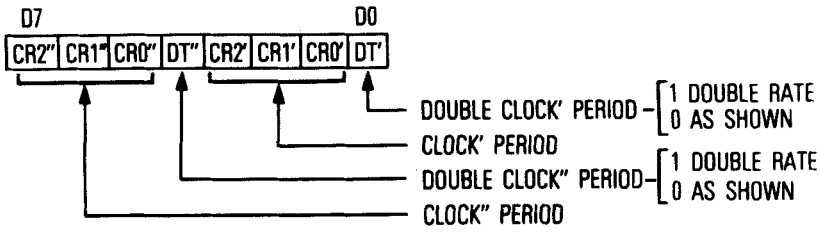

FIGURE 2

The Clock Select Register (see Diagram 1b for its relation with other components). The clock frequency for timers 1 and 2 are determined by the Clock Select Register.

timed intervals with the least redundancy, while the whole number increments simplify selection. The first bit of each 4-bit Clock Select Register, D_1 and D_4, is a "double time" bit. When it is set high, the clock periods of Figure 3 are doubled. This greatly increases the set of intervals which can be accurately timed by adding a large number of even intervals which cannot be timed with the odd prime clock periods.

b. *Clock synchronization.* In addition, the two clock generators contain a "restart" (or reset) provision. The restart is triggered by a "Sync Pulse" generated when the gate for the timer goes high. Consequently, when the gate for Timer 1 or Timer 2 goes high, the timer is started and the corresponding clock is reset to the beginning of the clock cycle. This synchronizes not only the times but also their clocks and eliminates the possibility of starting the timers in the midst of a clock cycle. Thus the total error in any timed interval is reduced to less than one microsecond for both the normal frequency and the "Double Time" frequency.

	CLOCK PERIOD IN MICROSEC.												
	1	3	5	7	11	13	17	19	23	29	31	37	41
CR0'	0	1	0	1	0	1	0	1					
CR1'	0	0	1	1	0	0	1	1					
CR2'	0	0	0	0	1	1	1	1					
CR0"	0			1		0			1	0	1	0	1
CR1"	0			0		1			1	0	0	1	1
CR2"	0			0		0			0	1	1	1	1

FIGURE 3

A tabulation of clock rate codes indicating the clock periods and corresponding three bit code for each timer. The periods are prime numbers listed in microseconds. Setting of the "double time" bit doubles these clock periods.

3. The Timer Status Register.

The third register associated with the 8253 I.T. is the "Rimer Status Register" (see Diagram 1a). This is an input register which allows the microprocessor to "poll" status of timer outputs, gates, and the External Trigger Acknowledge line as shown in Figure 4. The data are latched into this register upon addressing it with an "input" instruction to insure current data. This register is particularly useful in verifying the status of the aforementioned outputs and lines and determining which half of the count a timer is on by polling the status of its output. This is necessary when the timers are used in a "stopwatch" manner (1) to increase the accuracy of the measured time.

4. The 8259 Interrupt Controller.

The 8259 Interrupt Controller (I.C.) is interfaced as shown in Diagram 1b. There are no auxillary registers associated with

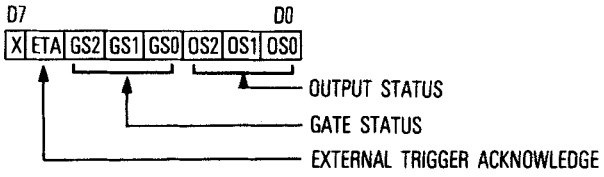

FIGURE 4

The Timer Status Register (see Diagram 1a for its relation with the interval timer, 8253). The microprocessor "polls" status of timer outputs, gates and external trigger acknowledge line.

it. The outputs of Timers 0, 1, and 2; and the external Trigger Acknowledge line are connected directly to the "Interrupt Request" lines (IRQ) 1, 2, 3, and 4, respectively. (IRQ 0 is being used by an A/D Converter in our system.) Consequently, anyone of these lines can initiate an interrupt, provided that the corresponding IRQ is unmasked (2). If this is done, a software subroutine can be synchronized with the timers or external trigger.

The programming of the 8259 I.C. is done with "input" instructions to read its output registers and "output" instructions to load its input registers. This masks and unmasks IRQ lines, determines the memory address vectored to upon receipt of an interrupt request, and determines the mode of operation (2).

The three "Cascade Inputs" (CAS 1, 2, and 3) and the $\overline{SP}/\overline{EN}$ output are not used. They are left available for cascading of three more 8259 Interrupt Controllers with the remaining three IRQ inputs, thus making it possible to add another 24 IRQ lines to the system (2).

ACKNOWLEDGMENTS

The authors thank Drs. W. R. Carper and R. V. Christian, Jr., for discussions and useful suggestions. The research was funded by the Wichita State University.

REFERENCES

1. Component Data Catalog, Intel Corporation, No. 8253/8253-5: Programmable Interval Timer, Santa Clara, CA, Section 8, pages 61-66, (1981).

2. Component Data Catalog, Intel Corporation, No. 8259A/8259A-2/ 8259A-8: Programmable Interrupt Controller, Santa Clara, CA, Section 6, pages 123-136, (1981).