gri-909

real time clock manual

GRI Computer Corporation
320 NEEDHAM STREET, NEWTON, MASSACHUSETTS 02164

Price $3.75

GRI-909

REAL TIME CLOCK MANUAL

GRI Computer Corporation, 320 Needham Street, Newton, Massachusetts 02164

10-47-007
0171-500

GRI-909 REAL TIME CLOCK MANUAL

TABLE OF CONTENTS

CHAPTER ONE

REAL TIME CLOCK

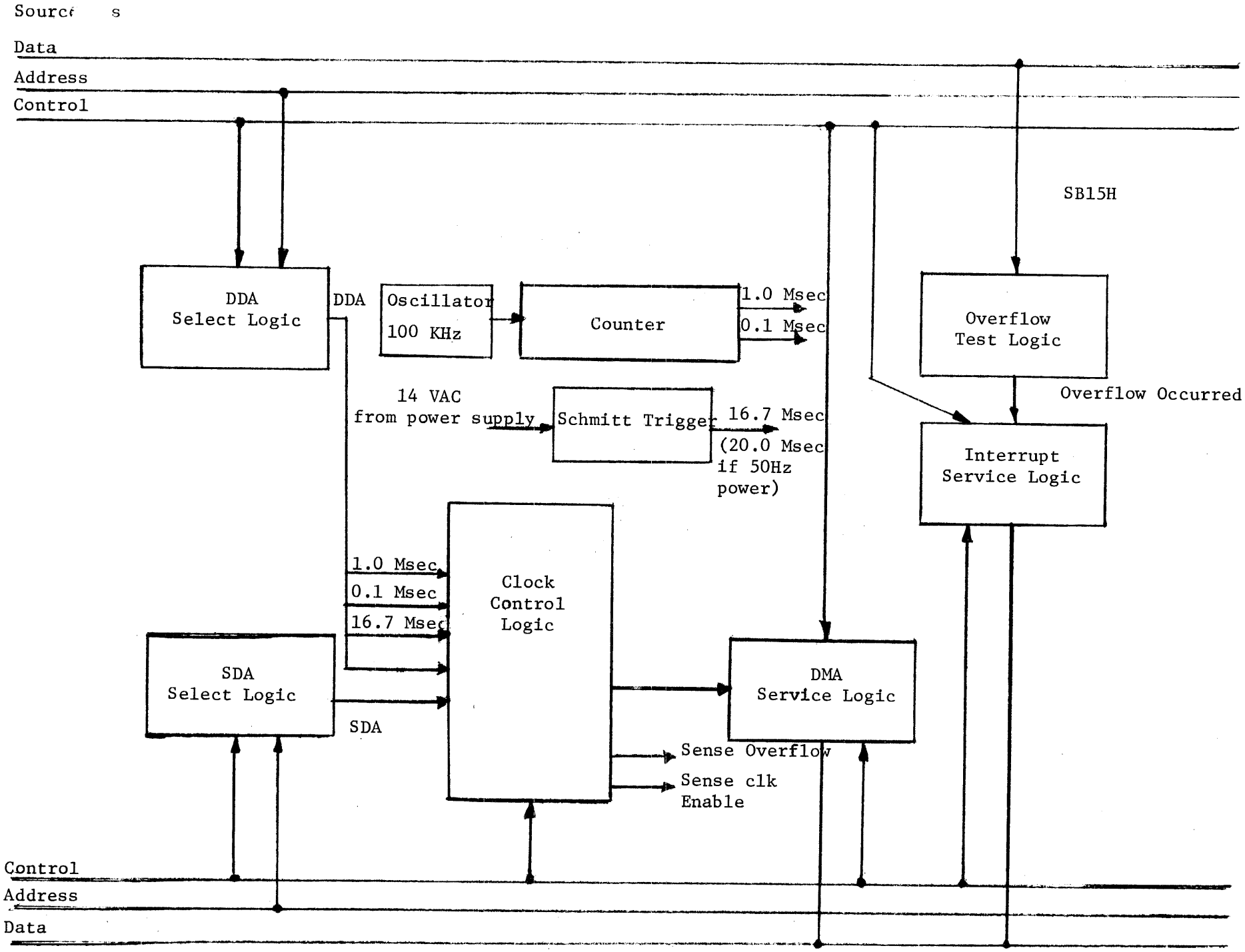
1.0 Introduction:

The GRI Real Time Clock is a firmware option that provides a "real" time base for the GRI-909 computers. Real time clocks are used to establish programmed variable time intervals for systems software. A real time clock could be used for a program that performs timed process control or simply for a program that must keep the time of day. A real time clock allows the computer program to keep in synchronism with a precision or external time base.

The GRI Real Time Clock is a direct memory access device that increments location 103 by one each time a pulse occurs. When location 103 overflows (increments from 177777 to 000000), an interrupt request is set. The clock will continue incrementing location 103 even after the interrupt request is set in case the program does not permit servicing of the clock before the next clock increment occurs. This feature allows the system to remain in synchronism with real time to a much higher degree of accuracy than turning the clock off after the overflow would allow.

By having the service routine pre-set location with some negative number, a time base which is any multiple of the basic clock frequency may be generated. For example, with a 60Hz clock, presetting location 103 to -73_8 (-59_{10}) generates an interrupt every second. Each time the interrupt occurs, the service routine would be entered and the clock location 103 pre-set to -73 again.

A block diagram of the GRI Real Time Clock is shown in Figure 1-1.



GRI Real Time Clock Block Diagram
Figure 1-1

1.1 Specifications:

Versions Available:

There are three versions of the Real Time Clock available:

1) The 60Hz Real Time Clock operates only at 16.67 ms (20 ms) time intervals, thus allowing the program to keep track of time in 1/60th of a second increments. The clock gets its timing from the 14v AC, 60Hz (50Hz) voltage source in the GRI-909 power supply.

2) The IC Oscillator Real Time Clock uses an integrated circuit oscillator to produce clock periods of 100 microseconds, and 1 millisecond. The integrated circuit oscillator operates at a 100KHz frequency, and by using two decade counters, the lower frequencies are produced. This version of the RTC will also operate at 60Hz.

3) The third version is identical to the 60Hz unit except it will accept an external time standard in place of the 60Hz source.

The frequency at which any RTC operates is determined by a wire jumper located on the back of the board by the external connector. (Location F2.) The clock can operate at only one fixed frequency at a time. To change frequencies, move the jumper from one location to another.

Clock Specifications (Other than 60/50Hz):

Operating Temperature ----- 0° to 50°C

DC Power ---------- 0.7 amps at +5v

AC Power ---------- 25MA at 14 VAC

IC Oscillator ---------- 100KHz ±0.005%

Stability ---------- 60 ppm for 0°C to 50°C

Aging ---------- 2 ppm/year

Input Pulse Specification:

The input pulse (for external time standard use) has the following specifications:

DC Voltage: Low 0v ±0.5v; High +18v ±2v at 70% high duty cycle max.

Min. pulse width = 100 N.S.

Max. frequency = 20KHz

AC Voltage: 11v AC to 14v AC RMS

1.2 Installation and Testing:

Installation:

The Real Time Clock is contained on one 9 X 4 inch I/O card that plugs into the back of the GRI-909 in any vacant slot. Its position relative to other devices will determine its priority during DMA and Interrupt operations. The order of priority is highest on the left side of the GRI-909 when facing the rear. Vacant slots between operators must be jumpered with pins (S40-215) in positions S-5, E and S-6, F. (Pages 2-16 and B-11 of the System Reference

Manual describe this in greater detail.)

The external connector, a 48 pin Amphenol (S40-203), is used to:

1. Bring 14v AC from the power supply Jones Strip to the 60Hz clock.
2. Make available to other system components the 3 oscillator frequencies. (Fan-Out of each frequency is 8 series 54/74 TTL loads.)
3. Accept an external time standard.

Preliminary Test:

To see that the RTC works sufficiently to run the diagnostic program, the following programs can be toggled into the computer and run to see that the indicated areas are working properly:

Check that RTC Runs:

To see that the RTC turns on and DMA's are serviced, run the following program. Make sure the clock is jumpered for either 100 or 1000 microseconds or the external source of clock pulses is properly connected.

Location	Machine Language	Function
0	02 0010 75	;Turn on RTC
1	00 0100 03	;Jump
2	000001	;To Self

When this program is started, the Memory Buffer will start incrementing through all 16 bits. Eventually, the MB should overflow and the MB should start counting from zero again. To stop the program, depress the stop key.

Check that Interrupt Works:

To see that the interrupt system works and that the RTC interrupts on overflow of Loc. 103, run the following program:

Location	Machine Language	Function
100	XXXXXX	;C(SC) saved here on interrupt
101	00 0100 03	;Jump To
102	000300	;Service routine
200	06 0010 04	;Set Int
201	004000	;Status Bit
202	02 1010 75	;CLR FLG, Turn on RTC
203	02 0010 04	;Turn on Interrupt
204	00 0100 03	;No Op Jump
205	000204	;To Self
300	75 1000 02	;Skip if OVFL SET
301	02 0100 00	;No
302	02 0100 00	;HALT
303	75 0100 02	;Skip if CLK EN SET
304	02 0100 00	;No
305	02 0100 00	;HALT
306	02 0001 75	;CLR CLK EN
307	06 0000 11	;Put DMA Counter
310	000103	;Into AX

```
311          11 0110 03        ;Jump is AX ≠ 0
312            000317          ;To HALT
313          02 1010 75        ;CLR FLG, Turn on RTC
314          02 0010 04        ;Turn on Interrupt
315          06 0000 07        ;Go Back
316            000100          ;To Program
317          02 0100 00        ;HALT; AX ≠ 0
```

After starting this program in Loc. 200, the program will continue to run if the clock is working correctly, until the stop key is pressed.

Real Time Clock Diagnostic Program:

Detailed testing of all functions and clock accuracy is done with DRT1 (79-43-010), a diagnostic routine for the Real Time Clock.

1.3 Programming and Operating:

The Real Time Clock can be broken up into four sections, as shown in Figure 1-1. They consist of the Clock Control, Clock, Direct Memory Access Service and Overflow Test, and Interrupt Service.

1.3.1 Clock Control:

This block contains the Device Decoding, Sense Function gates, and Function Output flip flops.

By issuing the correct address (75_8) and function test (FTB) or control bits (CB), the program turns on or off the clock and the program can sense the state of the Clock Enable and Overflow flip flops.

Function Output Instructions:

BASE ASSEMBLY LANGUAGE	FAST ASSEMBLY LANGUAGE	MACHINE LANGUAGE	FUNCTION
FO CLEAR,RTC	CLEAR TO RTC	02 0001 75	;Clear clock enable
FO SET,RTC	SET TO RTC	02 0010 75	;Set clock enable
FO CLOVF,RTC	CLOVF TO RTC	02 1000 75	;Clear overflow flag
The CB bits may also be micro-programmed, e.g.			
FO CLOVF,SET,RTC	CLOVF SET TO RTC	02 1010 75	;Clear overflow flag and set clock enable

Sense Function Instructions:

There are two functions that can be tested under program control as follows:

Clock Enable ---------- is true if the RTC has been turned on and is running.

OVERFLOW ---------- is true when the RTC has sensed an overflow in the counter at memory location 103.

BASE	FAST	MACHINE	FUNCTION
SF RTC,ON	SKIP IF RTC ON	75 0100 02	;Skip if clock is on
SF RTC,NOT ON	SKIP IF RTC NOT ON	75 0101 02	;Skip if clock is not on
SF RTC,OVFL	SKIP IF RTC OVFL	75 1000 02	;Skip if clock overflow is true
SF RTC,NOT OVFL	SKIP IF RTC NOT OVFL	75 1001 02	;Skip if clock overflow is not true

Note: The foregoing statements will assemble as shown provided they are preceded in the user program by the definitions. The definitions shown are recommendations for symbol assignments. The user may, of course, also assign his own mnemonics.

BASE	FAST
RTC=75	RTC #1,75
CLEAR=1	CLEAR #2,100
SET=2	SET #2,200
CLOVF=10	CLOVF #2,1000
ON=4	ON #3,400
OVFL=10	OVFL #3,1000

1.3.2 Clock:

Once CLOCK Enable (CLK EN) is turned on, the clock pulses from the 60Hz (50Hz) Schmitt Trigger, or one of the IC oscillator pulse trains (determined by the jumper on the board) causes a Direct Memory Access by setting the Clock Flag in the DMA service block. This in turn enables the setting of DMA Request at the next ISYNH time.

1.3.3 Direct Memory Access Service and OVFL Test:

Upon completing the next full instruction after the DMA Request is set, the processor goes into the DM state.

At T0 of this cycle, the processor requests the Memory Address of the location that the RTC will increment. This address is loaded into the Memory Address Register from the RTC when it sees the external address source address (EASH).

At T1 time of the DM cycle, the processor executes the micro-instruction MB TO EDDH. The Source Bus Data Lines will carry the previous value of location 103 during this time period. The value of SB15 is loaded into the OVERFLOW TEST flip flop during this time.

At T2 time the processor executes the micro-instruction MB P1. This increments the past value of location 103 in preparation for writing the new value back into location 103. The Source Bus line SB15 will now carry the new value of the clock word and pulse P2H is used to compare the contents of the OVERFLOW TEST flip flop with the new value of SB15.

This tells the RTC one of the following happened:

1. At T1 the contents of SB15 was a zero (OF Test not set), and at T2 C(SB15) is a one or zero, i.e. there was no counter overflow.
2. At T1 the C(SB15) was a one (OF Test is set), and at T2 C(SB15) is still a one, i.e. there was no counter overflow.
3. At T1 the C(SB15) was a one (OF Test is set), and at T2 C(SB15) is a zero. There was counter overflow, and the OVFL Flag is set.

The CLK Flag flip flop is cleared each time the processor is in a DMA cycle so that it is ready to be set on the next clock pulse and generate another request. This process continues until an overflow does exist. At that time, an Interrupt Request is initiated.

To vary the basic time interval between overflows (interrupts), the program can preset the DMA increment location (103_8) to the 2's complement of one less than the number of clock ticks wanted. For example: presetting Loc. 103 to 177400 and the

board jumpered for 100 microsecond clock; it will take 25.5 milliseconds for the clock to overflow. If the board was jumpered for 1 MS operation, the same number would take 256 milliseconds to overflow. If the board was set up for 60Hz operation, it would take 4.28 seconds to overflow.

The clock can be inaccurate as much as one clock tick when it is turned on. This is because the pulse sources are free running, and if the RTC is turned on just before the next pulse occurs, the time interval to the first pulse could be very short. A way to get around this problem is to preset Loc. 103 to -1 (all ones) and wait for the next clock pulse to cause overflow. At this time, the RTC can be turned back on, and the program will be in sync, within a couple of microseconds, with the RTC.

1.3.4 Interrupt Service:

As stated previously, an Interrupt Request is set when overflow occurs during a DMA cycle. This causes the processor to go directly from the DM state to service the Interrupt caused by the overflow without going back to execute an instruction which could take up to four machine cycles to finish. This reduces the possibility of the processor getting out of sync with the RTC.

The RTC Interrupt Status bit is bit 11. Thus, an instruction MRI 4000,ISR enables the RTC to interrupt when Overflow occurs.

An automatic trap jump to location 100_8 occurs where:

```
100        000000          ;C(SC) saved here on interrupt
101        06 0010 07      ;Jump to RTC
102        XXXXXX          ;Service routine address-1
103                        ;DMA increment location
```

The clock keeps running after Overflow has occurred and Interrupt Request is set. This is to allow the program to resync. itself to the RTC if the program is not able to service the Interrupt as soon as it occurs. For example, if the program calls for an External Instruction that takes many milliseconds and the RTC Overflows during the External Instruction, the program cannot service the Interrupt until after the EI is finished. By examining the contents of the DMA counter (103) during the service routine, the program can tell how many times the RTC ticked since Interrupt occurred, thus resyncing itself.

GRI Computer Corporation

320 NEEDHAM STREET, NEWTON, MASSACHUSETTS 02164

TEL: (617) 969-0800