SDM-INT8
8 CHANNEL INTERVAL TIMER
INSTRUCTION MANUAL

REVISION: 10/91

WARRANTY AND ASSISTANCE

The **SDM-INT8 INTERVAL TIMER** is warranted by CAMPBELL SCIENTIFIC, INC. to be free from defects in materials and workmanship under normal use and service for twelve (12) months from date of shipment unless specified otherwise. Batteries have no warranty. CAMPBELL SCIENTIFIC, INC.'s obligation under this warranty is limited to repairing or replacing (at CAMPBELL SCIENTIFIC, INC.'s option) defective products. The customer shall assume all costs of removing, reinstalling, and shipping defective products to CAMPBELL SCIENTIFIC, INC. CAMPBELL SCIENTIFIC, INC. will return such products by surface carrier prepaid. This warranty shall not apply to any CAMPBELL SCIENTIFIC, INC. products which have been subjected to modification, misuse, neglect, accidents of nature, or shipping damage. This warranty is in lieu of all other warranties, expressed or implied, including warranties of merchantability or fitness for a particular purpose. CAMPBELL SCIENTIFIC, INC. is not liable for special, indirect, incidental, or consequential damages.

Products may not be returned without prior authorization. To obtain a Returned Materials Authorization (RMA), contact CAMPBELL SCIENTIFIC, INC., phone (435) 753-2342. After an applications engineer determines the nature of the problem, an RMA number will be issued. Please write this number clearly on the outside of the shipping container. CAMPBELL SCIENTIFIC's shipping address is:

CAMPBELL SCIENTIFIC, INC.
RMA#_____
815 West 1800 North
Logan, Utah 84321-1784

CAMPBELL SCIENTIFIC, INC. does not accept collect calls.

Non-warranty products returned for repair should be accompanied by a purchase order to cover the repair.

CAMPBELL SCIENTIFIC, INC.

815 W. 1800 N.
Logan, UT 84321-1784
USA
Phone (435) 753-2342
FAX (435) 750-9540
www.campbellsci.com

Campbell Scientific Canada Corp.
11564 -149th Street
Edmonton, Alberta T5M 1W7
CANADA
Phone (403) 454-2505
FAX (403) 454-2655

Campbell Scientific Ltd.
Campbell Park
80 Hathern Road
Shepshed, Leics. LE12 9RP
ENGLAND
Phone (44)-50960-1141
FAX (44)-50960-1091

SDM-INT8
TABLE OF CONTENTS

APPENDICES

LIST OF FIGURES

LIST OF TABLES

PAGE

SDM-INT8 8 CHANNEL INTERVAL TIMER

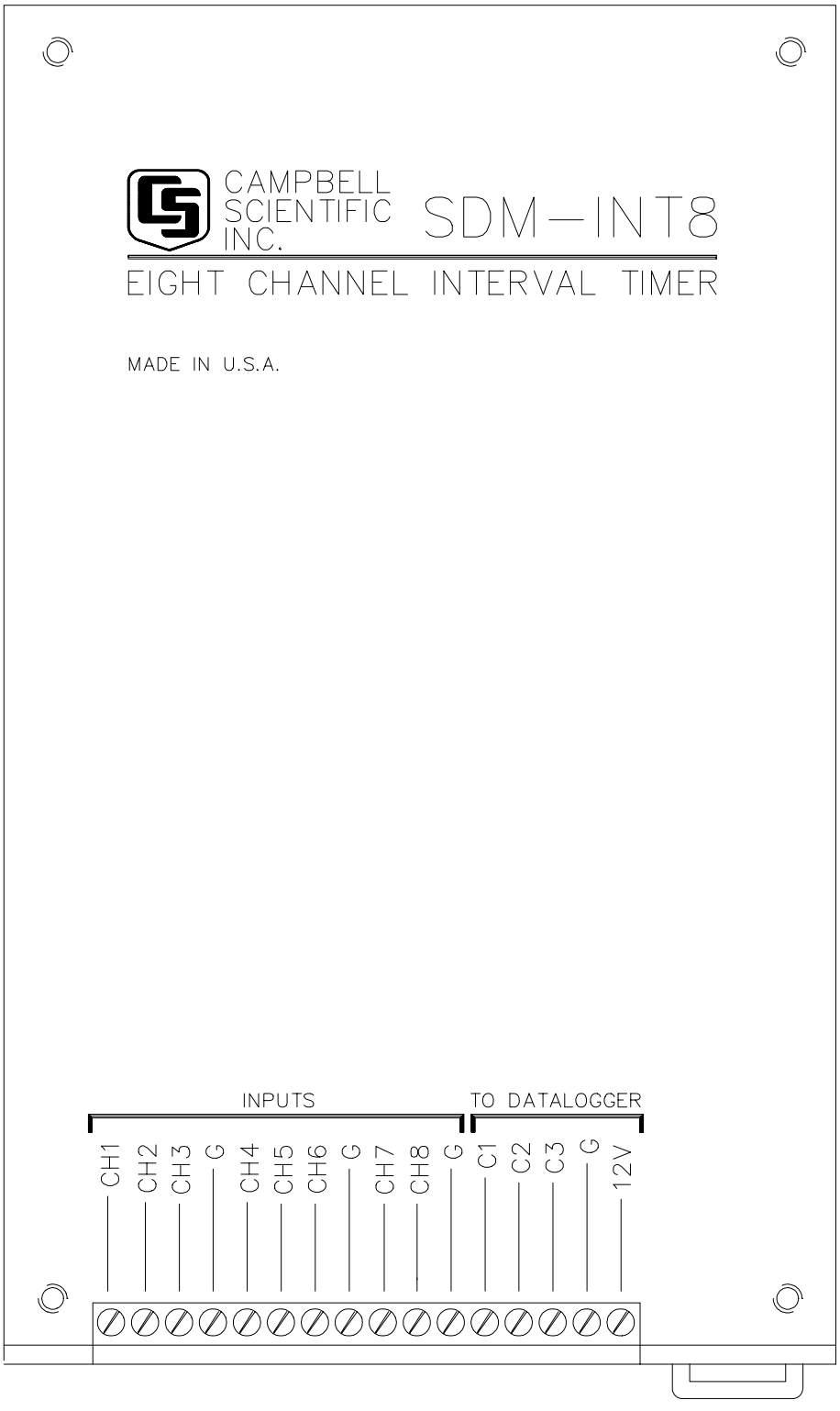

FIGURE 1. SDM-INT8 Front Panel

1. OVERVIEW

The 8 channel Interval Timer (INT8; see Figure 1) is a measurement module which outputs processed timing information to a CR10 or 21X datalogger. Each input channel is programmed to detect transitions from low or high level voltage inputs (Figure 2). Period, pulse width, frequency, counts, and time intervals are output to the datalogger for further processing/logging.

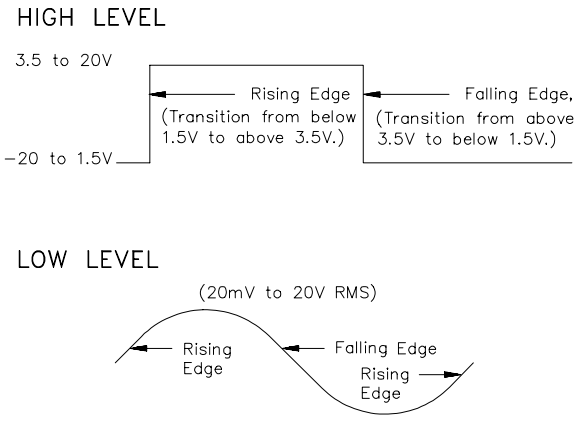

FIGURE 2. Voltage Input Options, Edges, and Thresholds

Datalogger Instruction 101 addresses, commands, and receives data from the INT8 through Control Ports 1, 2, and 3. Multiple INT8s, each with a unique address, may be controlled by one datalogger.

The INT8 has its own processor which enables it to make measurements and process data while the datalogger is performing other tasks. Each of the 8 channels may be independently programmed to detect either rising or falling edges and perform the following functions (Section 5.3).

Function Number	Description
1	- period (ms)
2	- frequency (kHz)
3	- time of signal edge since signal edge on previous channel; "time since previous channel"
4	- time of signal edge since signal edge on channel 1; "time since channel 1"
5	- counts on channel 2 that occur between signal edge on channel 1 and signal edge on this channel; "counts on channel 2 since channel 1"
6	- low resolution frequency (kHz)
7	- counts since last execution of Instruction 101; "counts"
8	- integral counts on channel 2 that occur between signal edge on channel 1 and signal edge on this channel; "integral counts on channel 2 since channel 1"

Results from these functions can be sent to the datalogger as an average, or each event can be captured depending on the Output Option selected (Section 5.4).

Frequency (Functions 2 and 6) data may be in a high or low resolution format, using 32 or 16 bit values respectively. Low resolution has a faster response, requiring less time to transfer data to the datalogger.

The INT8 can capture timing events with 1 microsecond resolution over a maximum range of 16.77 seconds. Timing on different channels can be compared to within ± 1 microsecond. At the same time, the datalogger can be executing various analog measurements, but the exact time these measurements are taken is subject to the datalogger's timing resolution. Section 7.3 discusses the possibilities and limitations of synchronizing INT8 and datalogger measurements.

2. SPECIFICATIONS

Operating voltage: 9.6 V to 16 V DC

Current drain: 13 to 20 mA active, 400 microamp quiescent

Environmental: -25 to +50 degrees Celsius 0 to 90% RH (non-condensing)

Number of channels: 8

Maximum timing measurement: 16.7 seconds

Resolution: ± 1 microsecond

Size and weight: 8 X 5 X 1 inch, 1.8 lbs

Input voltage option per channel: high level, low level

High Level Voltage Input:

- Minimum pulse width: 2 microseconds
- Signals edges:
  rising: transition from < 1.5 to > 3.5 volts
  falling: transition from >3.5 to<1.5 volts
  Maximum input voltage: 20 volts
- Maximum frequency:
  5.1 kHz when using Averaging Options
  10 kHz when Capturing All Events

The Low Resolution Frequency function allows higher frequencies to be measured if it is used on all programmed channels with Execution Interval Averaging. Maximum frequency is dependent on the number of channels programmed, as shown below:

No. of Channels	Max Freq (kHz)
1	42.5
2	17.5
3	11.0
4	8.6
5	5.2
6	4.8
7	4.5
8	4.28

Low Level Voltage Input:

- Minimum AC voltage: 20 millivolts RMS
- Input Hysteresis: 11 millivolts
- Maximum AC voltage: 20 volts RMS
- Minimum frequency: 1 Hz
- Maximum frequency:

Minimum AC voltage RMS	Max Freq. (Hz)
20 mV	100
50 mV	400
150 mV	1000
2.5 V - 20 V	4000

3. CONNECTIONS

Connections between the datalogger and the INT8 are shown in Figure 3. INT8s are shipped from the factory with a 10K Ohm resistor attached to the terminal strip for the convenience of 21X user. This resistor is necessary only when the INT8 is used with a 21X datalogger.

CR10 - The order in which CR10/INT8 connections are made is critical. The CR10 cases and wiring panel bracket are at datalogger ground. To avoid accidentally shorting 12 V to the case, connect the 12 V first then the ground. To prevent voltages in excess of 5 V from entering the Control Ports (C1-C3), the Control Ports are wired after connecting the ground lead.

21X - The order in which 21X/INT8 connections are made is not critical. A 10K resistor is wired between Control Port 1 and single ended input 1 (1H).

The total cable length for all SDMs attached should not exceed 20 feet.

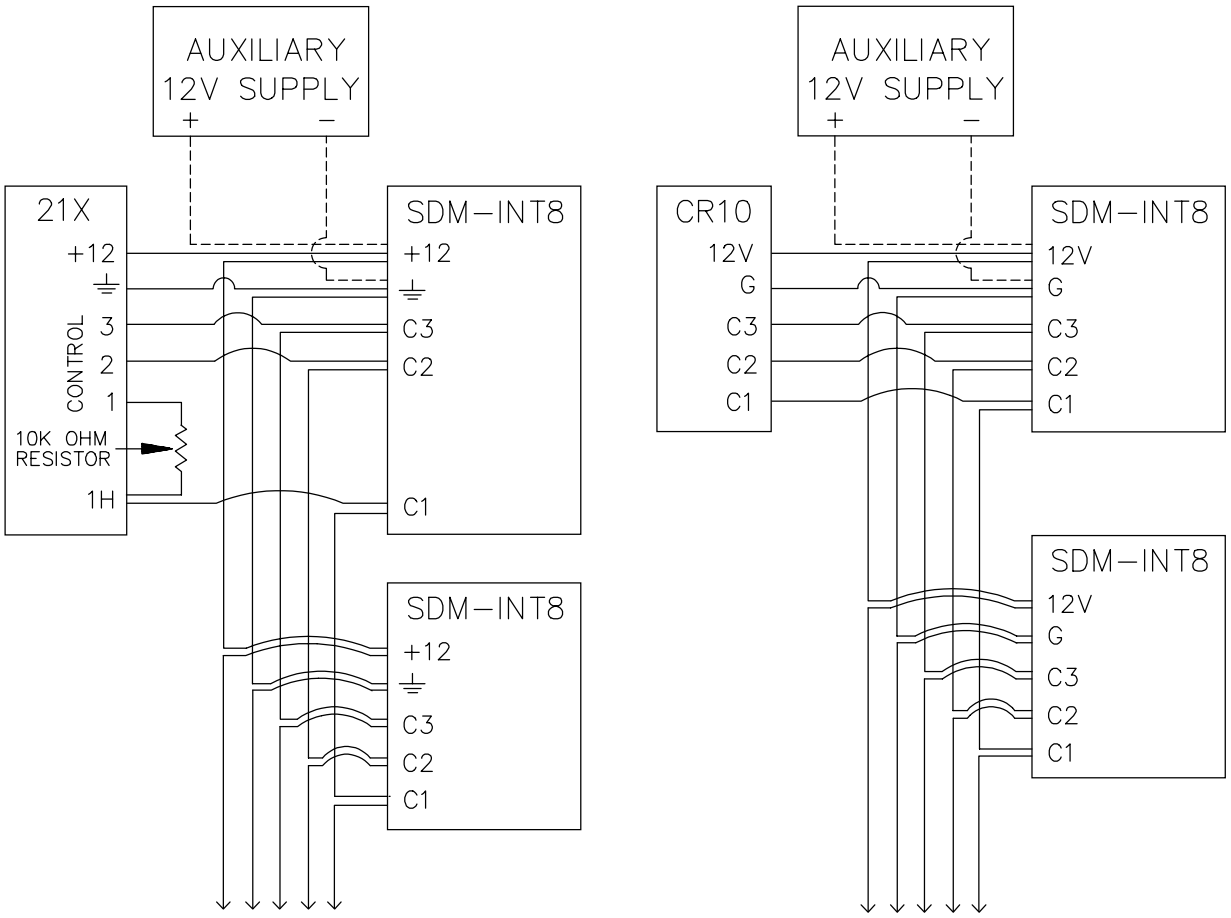

FIGURE 3. Wiring Diagram

4. POWER SUPPLY CONSIDERATIONS

The datalogger's power supply is typically used to power the INT8, however, an auxiliary supply may be used as shown in Figure 3. When selecting a power supply, consideration must be given to the active current drain and the active time of the INT8. With two exceptions, if the INT8 is programmed it is drawing 13 to 20 mA. The two exceptions are:

- When the Specified Averaging Interval (Output Option Section 5.4) is selected, the INT8 enters the quiescent current drain state (400 microamp current drain) after returning the results to the datalogger.
- If the interval between executions of Instruction 101 exceeds 16.77 seconds, the INT8 enters the quiescent current drain state.

If a 21X datalogger is used to power the INT8, all low level analog measurements (thermocouples, pyranometers, etc.) must be made differentially. This is due to slight shifts in the ground potential on the terminal strip when the 21X is used to power external devices.

5. PROGRAMMING THE DATALOGGER

Instruction 101 (see Table 1) is used to address, command, and retrieve data from the INT8. On the first execution of Instruction 101, the INT8 is programmed. Subsequent executions of Instruction 101 may command the INT8 to send its processed data to the datalogger or to reinitialize its measurement process. If multiple INT8s are connected to a datalogger, each INT8 must have a corresponding Instruction 101 and a unique address.

The datalogger tracks the first time a 101 Instruction is executed to know if the associated INT8 is programmed or not. If two or more 101 Instructions are used to address the same INT8, the first execution of each Instruction will program the INT8 returning no data. If the INT8 must be called more than once per datalogger execution interval, place Instruction 101 in a subroutine and call the subroutine when Instruction 101 must be executed. The INT8 is programmed on the first call, with data being returned on subsequent calls.

5.1 PARAMETER 1 - ADDRESS

The INT8 is enabled by an address sent from the datalogger. A terminal block located inside the INT8 has two jumpers which define the INT8 address. The address defined by the jumpers must match the address entered into parameter 1. The jumpers are set at the factory for address 00. If the jumpers have not been changed, the entry for parameter 1 is 00. Each INT8 connected to the datalogger must have a different address. See Appendix A for details on changing the address.

TABLE 1. Instruction 101

Parameter	Data Type	Description
01:	2	Address
02:	4	*Input config; channels 8,7,6,5
03:	4	*Input config; channels 4,3,2,1
04:	4	**Function; channels 8,7,6,5
05:	4	**Function; channels 4,3,2,1
06:	4	***Output option
07:	4	Loc
08:	FP	Mult
09:	FP	Offset

Execution time: 2.3 ms + 1.65 ms/value + averaging interval
(See Appendix B to estimate processing time on higher frequency signals)
Intermediate Storage: 1 location
* Input configurations (Section 5.2):
0 = high level, rising edge
1 = high level, falling edge
2 = low level, rising edge
3 = low level, falling edge
**Functions (Sec. 5.3):
0 = no value returned
1 = period in ms
2 = frequency in kHz
3 = time since previous channel is ms
4 = time since channel 1 in ms
5 = counts on channel 2 since channel 1
6 = low resolution frequency in kHz
7 = counts
8 = integral counts on channel 2 since channel 1
*** Output (Sec. 5.4):

0	Execution interval averaging
0- -	Continuous averaging
XXXX	Specified averaging interval in ms, XXXX>0
XXXX- -	Capture all events until XXXX edges of channel 1 (0<XXXX<8000)
9999- -	Test memory

5.2 PARAMETERS 2 AND 3 - INPUT CONFIGURATION

Each of the 8 input channels can be configured for either high level or low level voltage input, and for rising or a falling edge detect (see Figure 2). One digit (0,1,2, or 3) is specified to configure each channel, as shown below.

0 = high level, rising edge
1 = high level, falling edge
2 = low level, rising
3 = low level, falling

Example:
Channel 1 thru 5: high level, rising edge
Channel 6: high level, falling edge
Channel 7 and 8: low level, rising

02:2210 (channels 8,7,6,5)
03:0000 (channels 4,3,2,1)

5.3 PARAMETERS 4 AND 5 - FUNCTIONS

Each of the 8 channels can be programmed independently for various timing functions. Channel functions are programmed with one digit (0,1,2,3,4,5,6,7, or 8) for each channel in parameters 4 and 5. Functions 0 through 8 are described below.

0 - no value

1 - Period (ms)--
The time between signal edges on this channel in milliseconds.

2 - Frequency (kHz)--
The frequency of signal edges on this channel in kHz. Frequency is calculated from a measurement of period.

3 - Time since previous channel (ms)--
The time between the signal edge on the next lower numbered channel and the signal edge on this channel is in milliseconds. This function can be used to measure pulse width by connecting the signal to two adjacent channels programmed with opposite edge detect directions.

4 - Time since channel 1 (ms)--
The time between the signal edge on channel 1 and signal edge on this channel is in milliseconds.

5 - Count on channel 2 since channel 1--
The number of signal edges on channel 2 between channel 1's signal edge and this channel's signal edge. Linear interpolation is used to derive a fraction of a count at both the beginning and end of counting

6 - Low resolution frequency (kHz)--
For this function to return low resolution data, it must be the only function used in the instruction. If this is not the only function used in the instruction, it returns high resolution data, the same as Function 2.

Higher frequencies may be measured if fewer channels are used (see Section 2, Specifications).

For this function, any data value less than 1 is returned as 0. This must be considered when calculating the multiplier and offset (Parameters 8 and 9).

The frequency is returned to the datalogger in a low resolution format (16 bit floating point). This format allows for a range of positive real numbers between 1 and 65480 with 4 digit resolution on values whose mantissa is less than 8192. Three (3) digit resolution is given on values with mantissas greater than or equal to 8192.

Fewer bits are transferred to the datalogger in the 16 bit format, speeding up the instruction execution time by 0.3 ms per value.

7 - Counts--
"Counts" will always return an integer value when Instruction 101 is executed. The value will be the number of edges that have occurred since the last execution of Instruction 101. If no edges have occurred, a zero is returned. This function does not work with the Capture All Events Output Option.

If counts are being totalized by the datalogger, use the Continuous Averaging Output with "Counts" to avoid missing any counts (Section 5.4).

8 - Integral counts on chnl 2 since chnl 1--
Same as function 5 with no linear interpolation.

Example: The INT8 is used in an automotive test to measure crank angle, engine RPM when spark #1 fires, and fuel injector duty cycle. Parameters 2 and 3 are programmed as follows:

Parameter 2:0001 (channels 8,7,6,5)
Parameter 3:0000 (channels 4,3,2,1)

Channel 1: crank shaft reference pulse, rising edge (0)
Channel 2 pulse from the flywheel teeth, rising edge (0)
Channel 3: pulse from spark #1, rising edge (0)
Channel 4: fuel injector pulse, rising edge (0)
Channel 5: fuel injector pulse, falling edge (1)

Parameters 4 and 5 are programmed for the following functions:

Parameter 4:0003 (channels 8,7,6,5)
Parameter 5:2502 (channels 4,3,2,1)

Channel 1: frequency (2)
Channel 3: counts on channel 2 since channel 1 (5)
Channel 4: frequency (2)
Channel 5: time since previous Channel (3)
Channel 2,6,7,8: none (0)

Channel 1: RPM may be calculated from crankshaft frequency.

Channel 3: Flywheel teeth count between the crankshaft reference pulse, and the spark gives reference to the crank angle.

Channel 4: The frequency of fuel injection may be multiplied by channel 5's positive pulse width to yield fuel injector duty cycle. The multiplication is not performed in the INT8.

Channel 5: Time of the positive pulse width of the fuel injector is given in milliseconds.

5.4 PARAMETER 6 - OUTPUT OPTION

An important conceptual difference between Output Options and Functions (Parameter 4 and 5) is that one Output Option is selected per Instruction 101 and applied to data from all channels. Functions are applied to individual channels. The Output Options are as follows:

0	Average over execution interval
0--	Continuous averaging
XXXX	Averaging over specified interval
XXXX--	Capture all events (no averaging)
9999--	Test memory

5.4.1 Rules for Averaging

This Section applies to only those Output Options which perform averaging (0, 0--, and XXXX).

Averaging is performed on events which are defined by at least two edges. For example, to average a period two rising edges are required to define a period. To average the time since the previous channel, an edge on the previous channel followed by an edge of the channel programmed for time since previous channel is required.

No averaging is done on channels programmed for Function 7, "Counts". Channels programmed for counts do not require two edges for an event. Single edges of the specified direction are counted.

The maximum interval that the INT8 can time is 16.77 seconds. Edges which are separated by a time longer than this will result in a false measurement.

In all Output Options that average, the INT8 is storing measurements and processing. The measuring/storing task takes priority over the processing task. If the input signal exceeds a certain frequency, processing will lag behind measuring/storing. Table 2 provides the maximum average frequency at which the processing task keeps up with the measuring/storing task.

TABLE 2. Input Frequency (kHz) at Which Processing Time Equals Measuring/Storing Time

N	1,2,6,7	3	4	5	8
1	*	3.4	3.7	0.14 - 0.05F_2	3.4 - 0.81F_2
2	3.5	1.6	1.7	0.07 - 0.02F2	1.6 - 0.38F2
3	2.1	0.99	1.1	0.05 - 0.02F2	0.99 - 0.24F2
4	1.4	0.70	0.76	0.04 - 0.01F2	0.70 - 0.17F2
5	.99	0.53	0.57	0.03 - 0.01F2	0.53 - 0.13F2
6	.75	0.42	0.45	0.02 - 0.01F2	0.42 - 0.10F2
7	.59	0.34	0.37		
8	.47	0.28	0.30		

N = Number of channels measuring given Function
* = Greater than the maximum input frequency of 5.1 kHz
F_2 = Average input frequency on channel 2

See Appendix B to formulate the equations used to generate Table 2. Frequencies show in Table 2 are for "worst case" conditions. Faster input frequencies are possible depending on the phase relationship of the channel to channel signal.

With Options 0 and 0--, the average returned to the datalogger is the most recently processed average when the INT8 is addressed. If processing lags measuring/storing, the number of samples used in the average is reduced as is the effective averaging interval. For functions that average, this is not a problem, assuming the input frequency does not change significantly over the sampling interval. It is a problem if counts are being totalized (Function 7, Output Option 0--). In this case the count will intermittently be low (Section 5.4.3).

The Specified Averaging Interval Option (XXXX) uses all events captured over the specified interval to calculate an average. If the processing tasks gets behind the measuring/storing task, the additional time required to process all the edges is taken at the expense of Instruction 101 execution time (refer to Table 2.)

Due to finite memory in the INT8, when processing lags behind by 800 edges, the measuring/storing task is suspended for that interval. For Option XXXX to average over the entire specified interval, the interval must be short enough to prevent the processing tasks from getting behind by more than 8000 edges. Table 3 gives the sampling interval at which 8000 unprocessed events will accumulate for a given input frequency.

TABLE 3. Sampling Interval (Seconds) to Accumulate 8000 Unprocessed Events for Functions 1,2,6,7

Input	Number of Channels					
Freq. kHz	3	4	5	6	7	8
1.1						5.26
1.3				7.97	2.84	1.48
1.5				4.21	1.81	1.00
1.7				2.67	1.26	.73
1.9			5.62	1.87.	.93	.55
2.1			3.68	1.38	.72	.43
2.3			2.64	1.07	.57	.35
2.5			2.00	.86	.47	.29
2.7		6.55	1.57	.7	.39	.24
2.9		4.64	1.27	.58	.33	.20
3.1		3.50	1.05	.49	.28	.18
3.3		2.76	.88	.42	.24	.15
3.5		2.24	.75	.37	.21	.13
3.7		1.86	.65	.32	.19	.12
3.9		1.57	.57	.28	.17	.11
4.1		1.35	.50	.25	.15	.10
4.3	8.91	1.17	.45	.23	.13	.09
4.5	6.73	1.03	.40	.20	.12	.08
4.7	5.33	.91	.36	.19	.11	.07
4.9	4.36	.81	.33	.17	.1	.06
5.1	3.65	.73	.30	.15	.09	.06

Sampling intervals shown in Table 3 are for "worst case" conditions. Longer sampling intervals are possible depending on channel phase relationships. See Appendix B to calculate maximum intervals for other Functions.

When the low resolution frequency function is used, summing is not required; an average over the full interval is always available.

In all options that do averaging, the functions that involve time or count differences between different channels (time since channel 1, counts on 2 since 1, and time since previous channel) behave as follows: if there are multiple beginning edges and/or ending edges, i.e., more than 1 beginning edge per ending edge or visa versa, then only the last of the beginning edges and the first of the ending edges are used in the average. For example, assume all detection is on rising edges, and the function is counts on channel 2 since 1 for channel 3. A diagram of this example is presented in Figure 4. Multiple beginning edges are shown on channel 1, and multiple ending edges on channel 3. In this example, only 2 edges are used in the average.

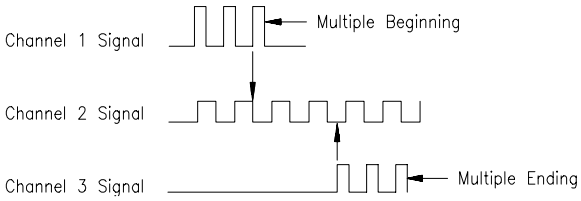

FIGURE 4. Example of Multiple Beginnings and Endings

5.4.2 Option 0: Execution Interval Averaging

When Option 0 is selected, the result from each channel is a value averaged over the interval since the INT8 was previously addressed. Normally this would be the execution interval of the Program Table containing Instruction 101. The value returned to the datalogger is the average at hand when the INT8 is addressed. If processing lags the measuring/storing task, the unprocessed events are not used in the averaging and are deleted from memory.

With Option 0, the datalogger should be programmed to execute Instruction 101 at least every 16.77 seconds while making measurements. If this period is exceeded, the INT8 enters a low power stand by mode and events are missed.

If no event occurs during the interval, 0 is returned for frequency and count functions, and 99999 (infinity) is returned for the other functions.

With Option 0, the INT8 ceases to capture events while it is communicating with the datalogger and reinitializes its measurement operation afterwards. Thus, edges that occur during this communication period (generally 2.3 ms + 1.7 ms/value) are ignored by the INT8.

5.4.3 Option 0--: Continuous Averaging

The Continuous Averaging Option (0--) is similar to Option 0 with the following differences:

1. The INT8 keeps capturing input edges during communication with the datalogger. If the time required for processing is less than or equal to the time required for measuring/storing (see Table 2), events will not be missed.

2. The datalogger will not update the input location of a channel that has not had an event since the last time the INT8 was addressed.

With Option 0-- the datalogger should be programmed to execute Instruction 101 at least every 16.77 seconds while making measurements. If this period is exceeded, the INT8 enters a low power stand by mode and events are missed.

Option 0-- is designed for input frequencies or intermittent signals that are at a slower rate than the execution interval of the datalogger. It is also used for totalizing counts.

Assume the input frequency is 0.5 Hz, and the execution interval of Instruction 101 is 10 Hz. The datalogger input location is updated at about 0.5 Hz if a change occurred, not every time the INT8 is addressed.

An example of an intermittent signal is a Cub Scout Pinewood Derby where the measured event is the elapsed time of a race, but several minutes separate each heat. Three cars per heat are gravity powered down a straight track. The start of the race triggers an edge on INT8 channel 1. The three cars each trigger an edge on separate channels as they cross the finish line. Function 4 is used to measure "time since channel 1" to provide the elapsed time for each car. The datalogger can be addressing the INT8 as often as desired with the 0-- option, but input locations will only be updated at the finish of a new heat.

The Continuous Averaging Option is used for totalizing counts (Function 7) because the INT8 continues to measure and store events even when communicating with the datalogger. However, counts will be missed if the measuring/storing task exceeds the processing task (see Table 2).

5.4.4 Option XXXX: Specified Averaging Interval

Option XXXX is used to average over an exact interval when Instruction 101 is executed. The averaging interval is specified in units of milliseconds. The datalogger program is delayed for the specified interval while the INT8 captures, edges, plus the extra time required for processing if the processing task lags behind the measuring/storing task. After returning the results to the datalogger, the INT8 enters the low power standby mode (400 microamp current drain), increasing battery life.

Like Option 0, this option will return 0 for frequency and count functions and 99999 for all functions if no result is available during the specified interval, i.e. the signal was too slow.

5.4.5 Option XXXX--: Capture All Events Until XXXX Edges On Channel 1

Option XXXX-- outputs all available timing information with no averaging. "All events" means that every occurrence of each programmed functions is recorded, i.e., each period, each pulse width, etc. as opposed to one averaged value.

Since the number of values returned may be variable, making it difficult to assign a fixed number of input locations, the Capture All Events Option outputs data directly to the datalogger's Final Storage when Instruction 101 is executed.

If the Output Flag is not set when the instruction is executed, the datalogger commands the INT8 to start measuring, and no results are returned. If the Output Flag is set, the datalogger retrieves the events that have been stored since the last time the instruction was executed and stores them directly in Final Storage. Data from the lowest numbered programmed channel are output first. Each subsequent channel's output has a new array ID which is incremented by 1. When all data are transferred, the INT8 starts its measuring process again.

The INT8 waits for the first edge on channel 1 as a "trigger" to start making measurements. It will then capture edges until XXXX edges on channel 1, or until the datalogger again addresses it, or until 8000 edges have been captured, whichever happens first. Output will not be returned until Instruction 101 is executed with the Output Flag set, even if the edge limits are reached.

With this option, Instruction 101 does not have to be executed every 16.77 seconds. The INT8 will continue to measure without entering the low power mode until 8000 edges are captured. Events being measured by the INT8 must not exceed 16.77 seconds, or false measurements will be returned. The maximum input is 10 kHz for all channels.

On a 10 second one-shot test, for example, Instruction 101 could be executed with the Output Flag cleared and then executed 10 seconds later with the Output Flag set. Or, Instruction 101 could be placed in a subroutine and called from different points in the programs.

For a 10 second test repeated every 10 seconds, the Output Flag could be set every time the instruction is executed.

Analog measurements can be made by the datalogger while the INT8 is capturing events. To some degree, and with caution, datalogger measurements can be synchronized with INT8 measurements. For example, the datalogger can execute the Burst Mode after executing Instruction 101 and trigger on the same signal that triggers channel 1 of the INT8. The caution is that the trigger on channel 1 cannot occur before the datalogger has a chance to enter the Burst Mode. To synchronize datalogger and INT8, the Burst Mode should be ready and waiting for the "trigger".

5.4.6 Option 9999--: Test Memory

Option 9999-- causes the INT8 to do a self memory test instead of measure and process timing functions. The signature of the INT8 PROM is returned to the datalogger's Input Location of the initial execution of Instruction 101. If the value is negative it indicates bad RAM; 0 indicates a bad PROM, as shown in Table 4.

TABLE 4. Definition of Test Memory Option Output

Output	Definition
positive integer	= ROM signature, good RAM
negative integer	= ROM signature, bad RAM
0	= bad ROM
-0	= bad ROM, bad RAM

A result of executing Option 9999-- is that the program residing in the INT8 is deleted, and the INT8 goes into the low current drain standby mode. To initiate INT8 measurements, Instruction 101 must be executed without 9999--.

5.5 PARAMETERS 7, 8, 9 - INPUT LOCATION, MULTIPLIER, AND OFFSET

Except for the Capture All Events Options, data from the INT8 are returned to the starting input location (parameter 7) in ascending programmed channel order. Output from Capture All Events is returned directly to Final Storage in the datalogger.

Except for the "Test memory" option, the multiplier and offset are applied to all results by the INT8 before they are returned to the datalogger.

For Low Resolution Frequency (Function 6), any value less then 1 is returned as 0. This must be considered when calculating the multiplier and offset.

6. OUTPUT FORMAT

The Output Option specifies the format of the output received by the datalogger. The format of the three options that do averaging (Options 0, 0--, and XXXX) are identical. A single value for each programmed channel is stored in consecutive input locations starting at the location specified in parameter 7, Instruction 101. If only three INT8 channels are programmed, then only three Input Locations will be utilized. Output Processing Instructions, such as Instruction 70 (Sample) must be used to store the results in Final Storage.

When using the Capture All Events Option (no averaging), the data are directed to Final Storage of the datalogger rather than Input locations. The first execution of Instruction 101 will program the INT8. Subsequent executions of Instruction 101 with the Output Flag set will output all events to Final Storage. If the Output Flag is not set, the INT8 will be reinitialized with out returning any data. The output returned for each programmed channel will be all events that have occurred since the last execution of Instruction 101. Each channel will have a unique Array ID. If other Output Processing Instructions precede Instruction 101, the first channel's data will be grouped with the previous Final Storage data (i.e. same Array ID.). This first Array ID will be incremented by one for each additional programmed channel.

Output for the Test Memory Option is a single value returned to the specified input location.

7. PROGRAM EXAMPLES

These examples are given to demonstrate concepts. The starting conditions for each example are followed by a datalogger program. These examples are not to be used verbatim. The programs are developed and formatted in EDLOG, a software module contained in Campbell Scientific's PC208 Datalogger Software Support package.

7.1 COLD CRANK ENGINE TEST - CAPTURE ALL EVENTS

The INT8 is used to obtain timing information during 20 crank cycles or 10 seconds of a 4 cylinder engine cold start test. A crank shaft reference pulse is wired into channel 1. Pulses from the flywheel enter channel 2 so that the counts on 2 since 1 function can be used to measure crank angle. The 4 spark plug firings are picked off a distributor wire and fed into channel 3. Channels 4 and 5 hook onto a fuel injector pulse for measurement of fuel injection pulse width.

The inputs into channels 1 - 5 are as follows:

Channel 1 - crank shaft reference pulse, rising edge
Channel 2 - pulse per flywheel tooth, rising edge
Channel 3 - pulses from the four spark plugs, rising edge
Signals common to single distributor pick-up wire
Channel 4 - fuel injector "on" pulse, rising edge
Channel 5 - fuel injector "on" pulse, falling edge

The channel functions are programmed as follows:

Channel 1 - Frequency, to get RPM
Channel 3 - Counts on 2 since 1, to get crank angle of the spark plug firings
Channel 5 - Time since previous channel to get pulse widths of the fuel injector.

Parameter 6 (Capture All Events) is programmed for 20--to capture edge times on channels 1-5 until 20 edges are received on channel 1.

Flag 2 is used in the example to indicate the start of the cranking. In actual test program, this flag could be set in response to an ignition switch measurement indicating the start of the test.

When the ignition is turned on (Flag 2 set), Instruction 101 is called, and the INT8 is programmed. A "loop with delay" is entered to delay for 10 seconds. The Output Flag is set high before executing Instruction 101 again to retrieve the test data.

Three arrays with the following Output array IDs are transferred to the datalogger's Final Storage.:

105 - Stored are hour:minute, seconds, channel 1 frequencies (kHz).
106 - Stored are crank angle results (teeth on flywheel).
107 - Stored are fuel injector pulse widths (ms).

Array 105 - Each time the crankshaft reference is reached, a pulse is generated on channel 1. INT8 measurements are started by the initial (trigger) pulse on channel 1. The pulse frequency (kHz) will be stored on each subsequent pulse on channel 1 (one pulse per revolution). The pulse frequency can be converted to RPM by multiplying by 6000. This conversion must be done in the computer, after the test.

Array 106 - Channel 2 receives one pulse each time a flywheel tooth passes a fixed reference point. When a crank shaft reference pulse occurs (channel 1), the crankshaft is in a known position. Each time a spark plug fires, a pulse occurs on channel 3. The output for channel 3 is the number of pulses that have occurred on channel 2 since a pulse from the crank shaft reference (channel 1). A value (number of pulses) will be output for each cylinder in their respective firing order. Every fourth data value in Array 106 will correspond to the same cylinder. The crank angle at each firing can be calculated by multiplying the number of teeth since the crankshaft reference pulse by 360/N, where N is the total number of teeth on the flywheel and subtracting this quantity from 360.

Array 107 - Channel 4 senses a rising edge when the fuel injector is activated. Channel 5 senses a falling edge when the fuel injector is deactivated. The INT8 calculates the time (ms) between these two edges, thereby calculating the pulse width, or the length of time the fuel injector is on for each pulse.

```
*      1                  Table 1 Programs
  01:              .5     Sec. Execution interval

01:    P91                If Flag                      If ignition is
  01:             22      2 is reset (Ignition not on) on, execute
  02:              0      Go to end of Program Table   following

02:    P86                Do                           address/prgm INT8
  01:              1      Call Subroutine 1

03:    P87                Beginning of Loop            loop to make a 10 sec.
  01:              1      Delay                        delay
  02:             20      Loop Count

04:    P95                End                          end of loop

05:    P86                Do                           set the output
  01:             10      Set high Flag 0 (output)     flag

06:    P77                Real Time                    time tag the output
  01:             11      Hour-Minute, Seconds

07:    P86                Do                           retrieve INT8 data
  01:              1      Call Subroutine 1

08:    P86                Do                           initialize flag 2
  01:             22      Set low Flag 2

08:    P                  End Table 1
```

```
*      3                Table 3 Subroutines

01:    P85              Beginning of Subroutine
  01:        1          Subroutine Number

02:    P101             SMD-INT8 (Extended)
  01:        0          Address Option
  02:     0001          Chan8765=HLrise/HLrise/HLrise/HLfall/
  03:     0000          Chan4321=HLrise/HLrise/HLrise/HLrise/
  04:     0003          Chan8765=none/none/none/TsinceP/
  05:     0502          Chan4321=none/C2sincel/none/kHz/
  06:      20--         Edges of Chan 1 in events mode
  07:        1          Loc: <--- Ignored in "Events" Option
  08:        1          Mult
  09:        0          Offset

03:    P95              End
```

7.2 WIND SPEED MEASUREMENTS - EXECUTION INTERVAL AVERAGING

The INT8 is used to make 1 second averages of frequencies from 6 RM Young Wind Monitors. Frequencies are converted to wind speed in m/s, and place in Input Locations 1 - 6. Wind directions are then measured, scaled, and placed in Input Locations 7-12. Programming does not output the data to the dataloggers' Final Storage.

Note that wind speeds of less than 0.2 m/s, corresponding to 2 Hz, will read 0 if a cycle of the Wind Monitor does not get completed within the 1 second execution interval.

```
*      1                Table  1 Programs
  01:        1          Sec. Execution interval

01:    P101             SDM-INT8                    measure wind speed
  01:       00          Address Option
  02:     0022          Chan8765=HLrise/HLrise/LLrise/LLrise/
  03:     2222          Chan4321=LLrise/LLrise/LLrise/LLrise
  04:     0022          Chan8765=none/none/kHz/kHz/
  05:     2222          Chan4321=kHz/kHz/kHz/kHz/
  06:        0          Execution interval averaging (1 second)
  07:        1          Loc [:m/s  ]
  08:     98.0          Mult m/s (convert from kHz to m/s)
  09:        0          Offset

02:    P4               Excite,Delay,Volt(SE)       measure wind direction
  01:        6          Reps
  02:       15          2500 mV fast Range
  03:        1          IN Chan
  04:        1          Excite all reps w/EXchan 1
  05:        0          Delay (units .01sec)
  06:     2500          mV Excitation
  07:        7          Loc [:degrees   ]
  08:     .1420         Mult degrees
  09:        0          Offset
```

7.3 BRAKE PRESSURE AND WHEEL SPEED TEST - CAPTURE ALL EVENTS AND P23 BURST MODE

A CR10 controls an INT8 to capture about 2 seconds worth of wheel speeds while measuring corresponding brake pressures in Burst Mode. The CR10's Burst Mode (P23) triggers the INT8 with Control Port 1 (C1) when the first analog channel measures pressure over the trigger level.

Flag 1 is manually set by the operator to control when the program is entered. After each test, the program sets Flag 1 low to return control to the operator.

The first time through the table, the INT8 is programmed and starts measuring, looking for the edge on channel 1 that the Burst Mode Instruction (P23) will trigger using C1.

The real time of the brake event (immediately after it occurred) is stored in array ID 9 with P77, followed by the analog burst data in arrays 1,2,3,4 and the timing data from the INT8 (picked up each time on the subsequent time through the table) in arrays 5,6,7,8.

The Burst Mode measures for 2 seconds (200 scans at 10 ms/scan). The INT8 measures for about the same 2 seconds, as defined by the trigger edge on channel 1 caused by the start of Burst Mode and the ending edge caused by pulsing Port 1 after the Burst Mode. Data from the INT8 is stored in Final Storage the *next* time the table is executed, and the INT8 is reinitialized for the next brake event.

```
*      1                    Table 1 Programs
  01:              .125     Sec. Execution interval

01:    P91                  If Flag/Port
  01:           21          Do if flag 1 is low
  02:            0          Go to end of Program Table

02:    P86                  Do
  01:           10          Set high Flag 0 (output)

03:    P80                  Set Active Storage Area
  01:            1          Final Storage Area 1
  02:            5          Array ID or location

04:    P101                 SDM-INT8
  01:           00          Address Option
  02:         0000          Chan8765=HL/rise/HLfall/HLrise/HLrise/
  03:         0000          Chan4321=HLrise/ HLfall/HLrise/HLrise/
  04:         0002          Chan8765=none/none/none/kHz/
  05:         2220          Chan4321=kHz/kHz/kHz/none/
  06:           1--         Edges of Chan 1 in events mode
  07:            1          Loc  :  (ignored in events mode)
  08:            1          Mult
  09:            0          Offset
```

05:	P23		Burst Measurement
01:		4	Reps
02:		12	7.5 mV fast Range
03:		1	IN Chan
04:		2100	TrigINchanSetC1/AboveLim/INpStr/Diff
05:		10	Time per (ms.)
06:		.2	Scans (in thousands)
07:		0	Samples before Trigger
08:		10	mV Limit
09:		0	mV Excitation
10:		1	Loc :
11:		1	Mult
12:		0	Offset
06:	P86		Do
01:		71	Pulse Port 1
07:	P86		Do
01:		10	Set high Flag 0 (output)
08:	P80		Set Active Storage Area
01:		1	Final Storage Area 1
01:		9	Array ID or location
09:	P77		Real Time
01:		11	Hour-Minute,Seconds
10:	P86		Do
01:		10	Set high Flag 0 (output)
11:	P80		Set Active Storage Area
01:		1	Final Storage Area 1
02:		1	Array ID or location
12:	P70		Sample
01:		200	Reps
02:		1	Loc
13:	P86		Do
01:		10	Set high Flag 0 (output)
14:	P80		Set Active Storage Area
01:		1	Final Storage Area 1
02:		2	Array ID or location
15:	P70		Sample
01:		200	Reps
02:		201	Loc
16:	P86		Do
01:		10	Set high Flag 0 (output)
17:	P80		Set Active Storage Area
01:		1	Final Storage Area 1
02:		3	Array ID location

```
18:   P70                Sample
  01:          200       Reps
  02:          401       Loc

19:   P86                Do
  01:           10       Set high Flag 0 (output)

20:   P80                Set Active Storage Area
  01:            1       Final Storage Area
  02:            4       Array ID or location

21:   P70                Sample
  01:          200       Reps
  02:          601       Loc

22:   P86                Do
  01:           21       Set low Flag 1

23:   P                  End of Table 1

*     A                  Mode 10 Memory Allocation
  01:          805       Input Locations
  02:           64       Intermediate Locations
  03:       0.0000       Final Storage Area 2
```

APPENDIX A. ADDRESS JUMPER

The address is factory set at address 00. Figure A1 (also shown inside the INT8 cover) shows the 16 possible positions and their corresponding address. Use the addresses shown in Figure A1 in Instruction 101.

The address block requires soldering to change the address. If two or more INT8s are to be used on the same datalogger, one can remain at address 00, while the others must be changed. Disconnect power before opening the case and soldering.

Every SDM device connected to the datalogger must have a unique address.

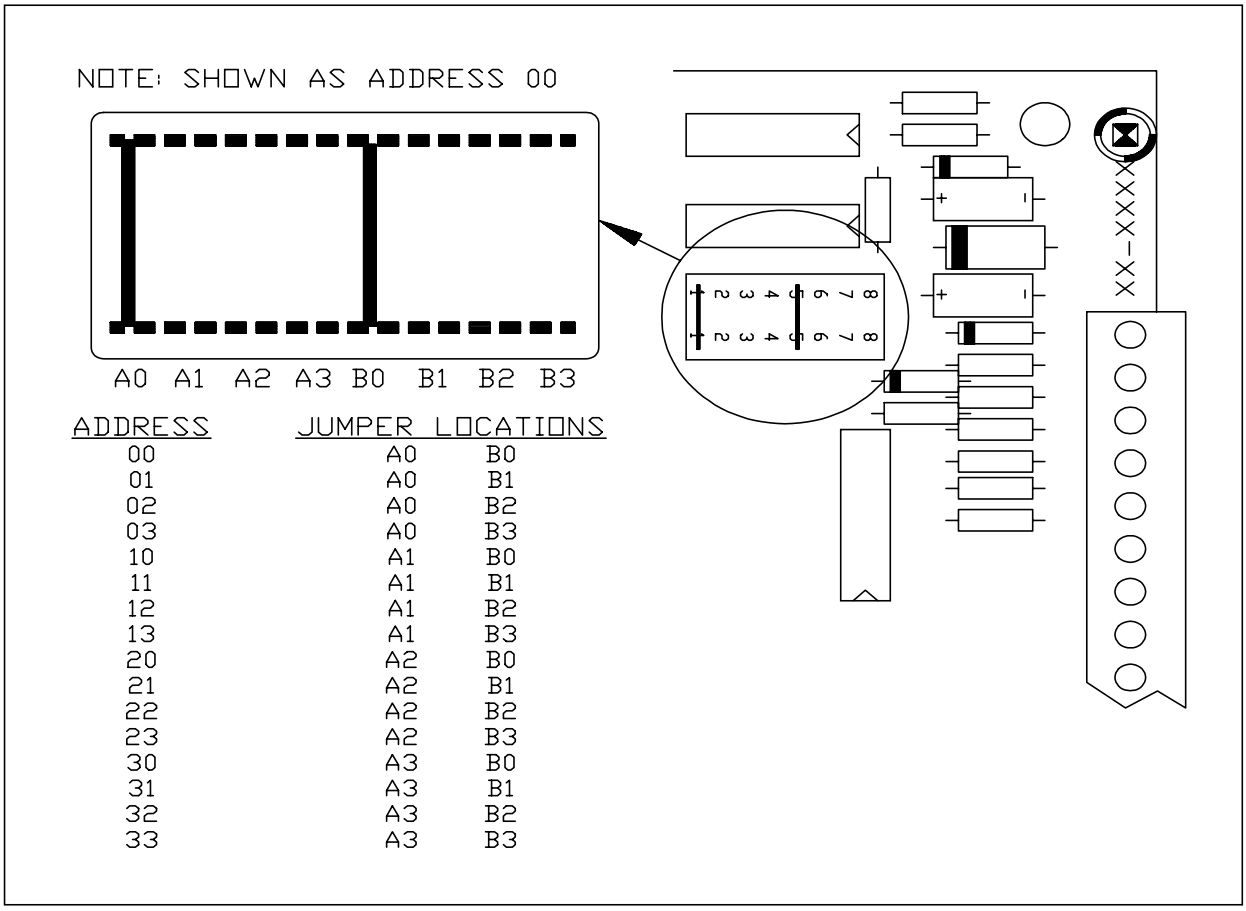

FIGURE A1. Address Selection

APPENDIX B. PROCESSING TIME LIMITATIONS

With the exception of the Capture All Events Option or when all channels are programmed for low resolution frequency (Function 6), the INT8 is performing two tasks simultaneously: (1) measuring and storing, and (2) doing the processing necessary for the final calculations of averages. The time required for processing varies depending on the total number of channels that are used, the total number of edges that occur during the averaging interval, and the function that each channel is programmed for.

With relatively high frequencies and particularly with the more involved processing of the Counts on 2 Since 1 Function, the processing task can lag behind the measuring and storing of the raw edge times. When averaging over the execution interval or when doing continuous averaging (Output Options 0 and 0--) this can shorten the actual averaging window. When averaging over a specified interval (Output Option XXXX) it can lengthen the execution time of the INT8 instruction. The following formulas are used to calculate the Total processing time (T_t).

$T_t = T_o + \Sigma T_i$

T_o = Sampling Interval Overhead, ms
$= 0.511*S + E_t*(.034+.010*N)$

Ti = processing Time for Function i, ms

$T_1 = 0.015*E_1$
$T_2 = 0.015*E_2$
$T_3 = 0.10*E_3$
$T_4 = 0.086*E_4$
$T_5 = 3.320*E_5 + .116*E_{c2}$
$T_6 = 0.015*E_6$
$T_7 = 0.015*E_7$
$T_8 = 0.1*E_8 + .116*E_{c2}$

where:

- N = Number of channels measured.
- E_i = Total number of edges occurring during the interval on those channels programmed with Function i (i = 1...8).
- E_t = Total number of edges during the interval on all channels.
- E_{c2} = Edges occurring during the interval on channel 2.
- S = The sampling interval in ms over which the INT8 samples the channels.

If the T_t for Output Options 0 and 0-- is greater than the sampling interval, the communication time with the INT8 will be extended from the minimum time (2.3 ms +1.65 ms/value) only by the processing time needed to obtain an answer for each channel, i.e., till at least one measurement is obtained. These Options should have data ready immediately except in the case of a very slow signal mixed in with relatively fast signals.

If the T_t for Output Option XXXX is greater than the specified interval, the extra time will be added to the execution time of Instruction 101.

Example 1
Given:
Output Option = XXXX
Specified interval = S = 500 ms
Function = frequency (Function 2)
Number of channels measured = N = 6
Average input frequency = F = 1kHz

Find: Total Processing Time, T_t

Solution:
$T_t = T_o + \Sigma T_i = T_o + T_2$
$= 0.511*S + E_t*(.034+.010*N) + 0.015*E_2$
$= 583$ ms

S = 500 ms
N = 6
E_t = N * S * F = 3000
E_2 = E_t = 3000

In this case 83 ms is added to the execution time of Instruction 101.

Example 2
Given: N = number of channels measured
F = Frequency of input signal, kHz
Function = frequency (Function 2)
Output Option = 0 or 0--

APPENDIX B. PROCESSING THE TIME LIMITATIONS

Find: Maximum average frequency, F, at which the processing time is ≤ the measuring/storing time.

Solution:

T_t must be ≤ to S;
$T_o+T_2 \leq S$

$0.511*S + E_t*(.034+.010*N) + 0.015*E_2 \leq S$

Note: $E_t = E_2 = N*F*S$

$0.511*S+N*F*S*(.034+.010*N)+$
$0.015*N*F*S \leq S$

$F<.489/(N*(.049+.01*N))$

Results for the above example using all Functions is presented in TABLE 2, Section 5.4.1

Example 3

Given: N = number of channels measured
F = Frequency of input signal, kHz
Function = frequency (Function 2)
Output Option = XXXX with specified sampling interval, S

Find: Sampling interval, S, such that the number of unprocessed events is ≤ 8000

Solution: The interval required to process 8000 events after the sampling interval is T_t-S, therefore;

$(T_t-S)*F*N \leq 8000$

$S \geq T_t-(8000/(F*N))$

$S \geq (0.511*S+E_t*(.034+.010*N)+0.015*E_2)$
$-(8000/(F*N))$

Note: $E_2 = E_t = N*F*S$

$S \leq 8000/[F*N((0.511+N*F*(0.49 +.01*N))-1)]$

Results for the above example are presented in Table 3, Section 5.4.1

APPENDIX C. INT8 INPUT SCHEMATIC