This is to certify that the

thesis entitled

A MICROPROCESSOR-BASED APPLE SCAB
DISEASE PREDICTOR

presented by

David Louis Neuder

has been accepted towards fulfillment
of the requirements for

Masters degree in Engineering

P. David Fisher
Major professor

Date May 18, 1979

A MICROPROCESSOR-BASED APPLE SCAB DISEASE PREDICTOR

By

David Louis Neuder

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Department of Electrical Engineering and Systems Science

1979

ABSTRACT

A MICROPROCESSOR-BASED APPLE SCAB
DISEASE PREDICTOR

By

David Louis Neuder

A microprocessor-based instrument predicts the growth and severity of the apple scab producing fungus Venturia inaequalis. Air temperature, relative humidity, rainfall, and leaf wetness serve as the weather data inputs to a table of growth curves that function as the basis for the prediction algorithm. The predicted scab severity level (1-of-4 levels), along with current date and time, sensor status, and eradicant spray recommendation, i.e., time left to apply any one of five different fungicides in order to avoid crop damage, is available via a keyboard and alphanumeric display. Also, a complete history of the sixteen most recent fungicidal infection periods and six most recent wetting periods is available for inspection. The instrument has been evaluated in the field and has proved to be extremely useful to both researchers and growers. Currently, the prediction algorithm is being reprogrammed to adapt the instrument to other pest management applications.

ACKNOWLEDGEMENTS

I would like to extend my sincere thanks to the following people for their supportive efforts in the project: Sig Lillevik, for consulting with me on the requirements of the ASDP, Tom Flynn and Kevin Pokallus for wiring the circuit boards of the ASDP, and Bonnie Beck for typing and proofreading the material. Furthermore, I would like to thank Dr. A.L. Jones for assisting me in the development of an apple scab prediction algorithm suitable for incorporation in the instrument. In addition, I would like to extend a special thanks to Dr. P.D. Fisher, whose constant encouragement and direction have made this work enjoyable and educational.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

The control of pest populations is an important aspect of agricultural production, forest management, and general improvement of human habitats. And pest management programs have long existed in a variety of forms. Unfortunately, the vast majority of these programs in well developed countries evolved to the point where there was excessive reliance on the use of chemicals for pest control. And, today, we know that many of these chemicals pose serious side-effects to both man and his environment. So, alternative pest management programs are being developed. Some rely on biological or natural control processes. Others recognize the long-term need for chemicals but are designed to reduce their use to a minimum.

The purpose of this work is to describe one such approach in which a microprocessor-based instrument alerts apple growers of pending apple scab infection periods. This instrument resides in an orchard and continuously monitors air temperature, relative humidity, rainfall, leaf wetness, time-of-day, and date. The grower uses a keyboard and a four-digit alphanumeric display to interrogate the instrument and determine if an infection period is likely to occur. Also, it informs the grower how many hours he has left to apply any one of five different fungicides on his crop in order to eradicate the apply scab fungus without incurring crop damage. In a normal year, this approach will reduce the grower's use of fungicides by 2-to-3 spray applications with reduction of 75 percent or more possible in some years, depending upon the weather. And this savings means that the average grower can recover his capital investment in the instrument within a couple of growing seasons.

This thesis presents the design and evaluation of this microprocessor-based approach. Chapter II details the design considerations and operational procedures that were incorporated into the instrument. Chapter III presents the hardware organization and implementation, including block diagrams of the major circuits, while the software organization and structure is included in Chapter IV. Chapter V summarizes the results and suggests further improvements for the instrument and this general approach to pest management.

CHAPTER II

PHYSICAL AND OPERATIONAL CONSIDERATIONS

This chapter establishes both the physical and operational requirements for the Apple Scab Disease Predictor (ASDP). The first section describes the physical considerations or general environmental and user oriented constraints that form the foundation of the instrument's design. The operational considerations, described in the succeeding four sections, then build upon these physical considerations to fully characterize the requirements of the ASDP. The first section of the operational considerations, the algorithm requirements, establishes the routine that must be implemented to predict apple scab infection. Following the prediction of an infection, the data storage requirement section specifies the amount of space that must be allotted to the recording of infection information. The operator interaction section then details the format and procedure in which stored data is to be displayed to the user. Finally, the debug interaction section specifies the debug functions that must be incorporated into the design to assist in hardware and software debug. Figure 2.1 details the interrelationship of the ASDP's requirements. In general, the design is an interactive process with knowledge gained while designing one aspect of the system affecting the design elsewhere.

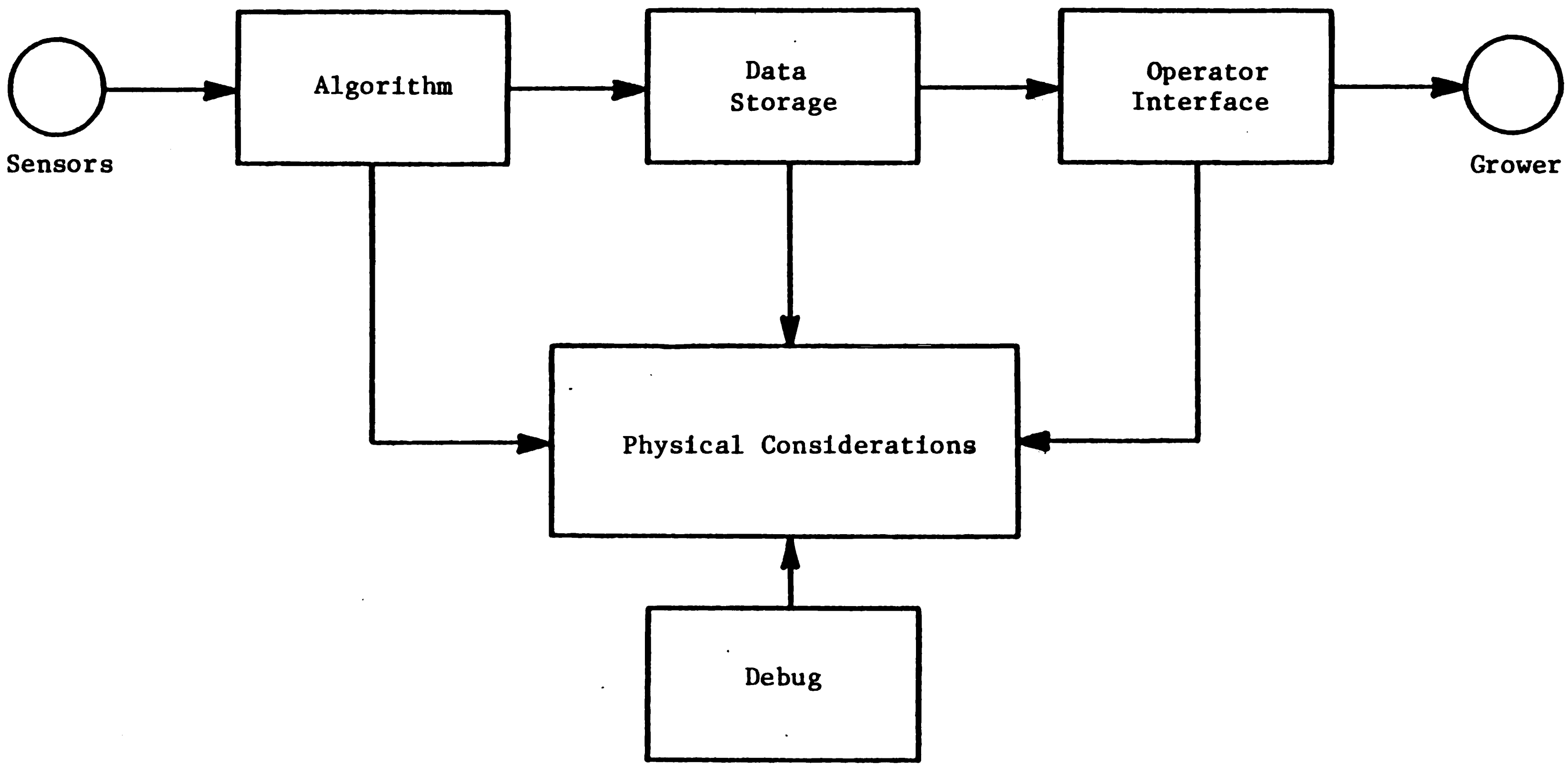


Figure 2.1 -- Interrelationship of Physical and Operational Considerations

2.1 Physical Considerations

In the development of an Apple Scab Disease Predictor numerous physical considerations must be taken into account. First of all, the unit must be designed to be mounted on a post at a representative location in an orchard. This location must provide the natural meteorological conditions that lead to accurate scab infection information. On the other hand, this location may not provide the environmental protection an unshielded instrument requires. Therefore, the instrument must be designed to operate through the various conditions of an uncontrolled environment. These may include: all possible atmospheric temperature and humidity conditions; dust and snow storms; rain, hail, or sleet; lightning or RF interference from farm machinery. In addition, a representative field location may not provide adequate power support for an instrument. Therefore, the unit must be battery powered, preferably with a battery supply capable of powering the instrument through the entire growing season from early spring to late fall. Second of all, utility to the apple grower must be considered in the design of this instrument. Many of the features that a grower would look for in purchasing heavy duty equipment should be incorporated into the design. Typically these features include: easy operation with minimal training; easy installation; minimal time commitment for routine maintenance; ruggedly built to withstand transportation forces; and use of inexpensive standard components throughout to facilitate repair if the need arises. In addition, the overall unit should be compact and inexpensive. These two conditions are necessities if the instrument is to be sold in numbers. Finally, the unit should be easily reconfigured for other applications or adding new functions. This feature eliminates the capital investment involved in the purchase of an entirely new instrument when a new algorithm is developed or a different pest is to be recorded.

2.2 Algorithm Requirements

The apple scab prediction routine directs the operation of the ASDP through the computation of meteorological intervals and prediction of fungal development. The scab routine predicts apple scab infection via an evaluation of meteorological conditions. These infection periods are predicted at the time of estimated fungus exposure which occurs 9-17 days before visible scab lesions occur.[1] The degree of scab infection predicted depends upon the amount of scab lesion formation expected. A predicted level of "low" scab infection indicates that scab development will most probably be light. Predicted scab severity levels include none, low, moderate, and high. Once the predicted level of scab severity is know, effective action (such as applying an appropriate spray) can be taken to reduce scab formation. But to determine a scab infection occurrence and the corresponding scab severity level the apple scab routine must monitor the environment to determine an infection period and an average air temperature during the infection period. The infection period and average air temperature are then utilized to determine a scab severity level via the modified Mills table (see Table I).[2] As an example: If an infection period listed nine hours with an average temperature of 69° F the predicted scab severity would be "low" and the expected crop damage would be light.

In brief summary, the apple scab routine consists of computation of an infection period, average air temperature, and the actual scab

TABLE I

MODIFIED MILLS TABLE

Average Air Temperature (°F)	Degree of Infection as a Function of Lapsed Time (hrs.) Low	Moderate	High
78	13	17	26
77	11	14	21
76	9½	12	19
63-75	9	12	18
62	9	12	19
61	9	13	20
60	9½	13	20
59	10	13	21
58	10	14	21
57	10	14	22
56	11	15	22
55	11	16	24
54	11½	16	24
53	12	17	25
52	12	18	26
51	13	18	27
50	14	19	29
49	14½	20	30
48	15	20	30
47	15	23	35
46	16	24	37
45	17	26	40
44	19	28	43
43	21	30	47
42	23	33	50
41	26	37	53
40	29	41	56
39	33	45	60
38	37	50	64
37	41	55	68
33-36	48	72	96

level. Computing an infection period is the major task for the apple scab routine. The infection period as listed by the Mills table consists of the number of hours in which the leaves are continuously wet. Nevertheless, a typical day may include wet, dry, and humid periods all within a few hours. In order to compensate for these various environmental effects, Jones has developed an algorithm to compute the infection period.[3] Through the use of this algorithm the infection period is composed of durations of time in which the climate is in one of the three states (see Table II).

The infection period begins with the occurrence of State A or wet leaves. No other state can start an infection period. All subsequent State A time durations and occurrences are summed into an ongoing infection period provided it has not been terminated. In this case, a new infection period will be started. Each occurrence of State A will extend an infection period and reset the termination period.

The termination period is the amount of accumulated time in State C (dry leaves, low humidity) which can be applied toward the termination requirements of the infection period. The predicted level of scab severity determines the termination requirements as indicated in Table III. Therefore, with low level scab severity, an accumulated termination period of six hours will curtail the infection period.

The duration of time in State B (dry leaves, high humidity) will only be accumulated into the infection period time if the predicted scab severity level is none. Once the none scab level has been passed, State B places the machine in a holding mode. In this mode State B will not be accumulated along with the infection period or termination period time. An infection period will not be extended or terminated.

TABLE II

INFECTION PERIOD STATES

State A Wet leaves	State A begins when 75% deflection of full scale is noted on the DeWit leaf wetness sensor. State A ends after deflection is within 10% of the dry resting position.
State B Dry leaves High Relative Humidity	Dry leaves occur whenver the leaves are not defined as wet. High humidity occurs when humidity is greater than or equal to 90%.
State C Dry leaves Low Relative Humidity	Dry leaves occur whenever the leaves are not defined as wet. Low humidity occurs when the humidity is less than 90%.

TABLE III

INFECTION PERIOD TERMINATION REQUIREMENTS

Predicted Scab Severity	Termination Requirements
None	8 continuous hours of State C
Low	6 hours of State C
Moderate	4 hours of State C
High	4 hours of State C

Contrary to terminating an infection period, the duration of time in State C (dry leaves, low humidity) can extend an infection period if all three of the following conditions are met: the predicted scab severity level is none; the continuous period of time in State C is preceeded and followed by States A or B; the time in State C does not extend beyond the termination requirements of eight hours. Once the predicted scab severity level has passed the none level, short periods of State C (less than the termination requirement) followed by States A or B will not extend the infection period. The example in Table IV illustrates usage of the Jones algorithm for computing the infection period. The average air temperature is computed utilizing data points throughout the total period of time in which the infection period occurs. These data points are recorded each half hour regardless of the climatic state or scab severity level. The mean of the stored data points then provides the apple scab routine with an average air temperature for the infection period.

The computation of the scab severity level as described earlier involves utilization of the infection period duration along with the average air temperature to address the modified Mills table and obtain a scab severity level. This severity level is then compared with earlier levels of the same infection period to determine the highest level of infection. The highest level is then selected as the new predicted scab severity level. If this selection process were not performed, a change in temperature could display a lower scab level when conditions had actually existed for a higher level infection to have occurred. In addition, the scab prediction routine operates every ten

TABLE IV

INFECTION PERIOD COMPUTATION

STATE	DURATION	INFECTION PERIOD	SCAB	TERMINATION PERIOD	COMMENTS
C	--	--	None	--	Initialization.
A	4	4	None	0	Wet leaves start infection period.
B	2	6	None	0	High humidity adds onto inf. period.
C	2	6	None	2	Dry leaves add onto termination period.
A	1	9	Low	0	Dry and wet leaves add onto inf. period. Termin. period is reset.
B	6	9	Low	0	State B no longer extends inf. period. State B places routine in hold mode.
C	4	9	Low	4	State C adds onto termin. period.
B	2	9	Low	4	State B places routine in hold mode.
A	5	14	Moderate	0	Only wet leaves add onto inf. period. Termination period reset.
C	1	14	Moderate	1	State C adds onto termination period.
B	2	14	Moderate	1	State B places routine in hold mode.
C	3	14	Moderate	4	Infection period terminates upon time reaching 4 hours.
C	--	--	None	--	Inf. period is reset.

32 hours = total duration

minutes to maintain a close contact with the environment and obtain an accurate prediction.

2.3 Date Storage Requirements

Accurate interpretation and evaluation of scab severity levels requires storage of relevant weather data and computed results associated with an infection period. Four storage blocks or buffers, each of which records specific types of data, comprise this storage function. The first of these, the full buffer, records weather data during the time period in which an infection occurs. This data includes: date and time of infection period start (first occurrence of wet leaves); average air temperature; total time during which infection period occurs; total times leaves were wet (total time in State A); total times leaves were dry/humidity low; total times leaves were dry/humidity high; and total rainfall. In addition, the scab prediction routine's computed level of scab severity is recorded in this buffer.

The full buffer is formatted in such a way that all of the information (9 total pieces)occupies 1/16 of the total buffer space. This allows 16 distant histories of scab infection to be held in storage. The earliest history is truncated upon the recording of a new infection period after the total buffer space is filled. Due to the fact that this buffer is tremendously important and storage space is limited, the buffer will only record infection periods greater than or equal to a five hours in duration.

Because weather data received during the first five hours of an infection is important in evaluation of instrument performance, a second buffer is required. The partial buffer fulfills the need by recording data from the start of an infection period until either a low level scab severity is reached or the infection period terminates. The partial buffer, like the full buffer, records the following data: date and time of infection period start (first occurrence of wet leaves); average air temperature; time leaves were wet; time leaves were dry, humidity low (State C time toward infection period); time leaves were dry, humidity high (State B time toward infection period); and rainfall. Unlike the full buffer, scab severity and total time of infection period are not recorded in the partial buffer and only six distinct history locations are available for storage of partial infection periods.

The partial and full history buffers specify a nearly complete history of any scab infection. A portion of the information which can be obtained from these buffers includes: infection period (by summing dry leaves, low, and high humidity from partial buffers with total wet leaves from full buffer); average temperature over infection period (directly from air temperature of full buffer); and time scab reaches low severity (add to start of infection time all time stored in partial buffer). One piece of information which is not currently stored in these history buffers is the step-by-step updating data. This data could force the scab severity to a higher level than anticipated by utilizing the Mills table with the average temperature and infection period as calculated from the history buffers. Therefore, not all of the information needed to characterize an infection period is stored in the history buffers.

Once a scab severity level has surpassed "none," action must be taken to protect the apple trees and crops. The only course of action is to apply eradicants (sprays which can destroy early fungal growth).

These sprays are limited in that they are effective for only a limited amount of time from the onset of infection. Therefore, they are effective for a correspondingly less amount of time once the scab severity level reaches a low. The third buffer's purpose then is to store these two times (onset of infection, low level scab severity) so that the remaining effective times of eradicants can be calculated. These times can then be presented to the user. This buffer functions identically to the history buffers except that only six date and time occurrences can be recorded (the equivalent of three infection periods).

The fourth buffer enables the user to obtain a crude estimate of the rainfall intensity by storing all occurrences of at least 3/100 of an inch of rain during a ten-minute interval. This information, when combined with other infection history data, can help to form a more concise picture by distinguishing between rainy and heavy dew periods. In addition, this buffer functions identical to the others except that only 12 date and time occurrences can be recorded.

2.4 Operator Interaction Requirements

The user-machine communication must provide the means with which apple scab predictions can be evaluated. Good communication revolves around a good user interface with a large battery of functions. The apple scab predictor-user interface centers on a ten digit two character telephone-type keyboard and four character alphanumeric display. The keyboard allows the user to input as well as request information. The display provides a means of visualizing the output data, as well as verifying the input data. The following contains a brief outline of how the keyboard-display interface works, and a description of the functions in detail.

The operation of the keyboard-display involves depressing a set-up ket (#), a function key (0-9), and a continue key (*). Depressing the set-up key (#) places the machine in the function select mode. In this mode, the next numerical key entered will select the function to be executed. One of the following functions may now be selected: calendar; scab; sensor; full history (down); full history (up); partial history; eradicant spray; prediction stop; rain density; and calendar set.

After the appropriate digit is entered, the continue key (*) should be used to step through the entire function. Upon completion of a function routine, the machine again enters the function select mode, where a new function can be immediately entered. Table V illustrates the proper operation of the calendar function. It outputs the current date and time.

TABLE V

KEYBOARD OPERATION EXAMPLE

Input Keyboard	Output Display
	OK
press #	+ (function select mode)
press 1	DATE
press *	JA22
press *	TIME
press *	1201
press *	+ (function select mode)

If the wrong key is pressed, one of the three following "error" messages will occur. The word ILGL (llegal) will appear anytime an illegal key has been pressed. This can occur if a key is pressed twice. The word INVD (invalid) will appear after an invalid password or pass number if entered. The word ERR will appear after an incorrect date or time has been entered. Removal from the error mode requires that key # be pressed. The # key takes presidence over all of the other keys and will return the machine to the function select mode. Three consecutive # key presses will return the display control to the machine. The display then circulates the characters *OK* under machine control. Moreover, the word POWR (power) will be displayed if the main power source is low.

In summary, the total operation of the keyboard-display interface revolves upon the selection of the proper key. Table VI lists all of the display functions and their descriptions.

2.5 Debug Interaction Requirements

The control and debug sections enable an operator to control and debug both hardware circuits and software routines. This type of control is required in a machine of this size to reduce trouble shooting time as well as new software development time. The nature of these sections require that they not be accessible to everyone. Therefore, they can only be activated upon opening the enclosure. The control section allows the user to specify the operational mode of the apple scab circuit. Varying operational modes may be obtained by using the control section switches listed in Table VII. Four of the most significant operational modes are: run, run fast, evaluate temperature

TABLE VI

KEYBOARD SELECTED DISPLAY FUNCTIONS

KEY		FUNCTION	
1		**Calendar Function** The calendar function displays current date and time. Date is listed as a two letter month followed by a two number day. Dates may range from Jan. 1 to Dec. 31. Time is listed in a 24 hour clock format.	
	Typical Output:	DATE FB03	Current date=February 3
		TIME 2312	Current time=11:12 p.m.
2		**Scab Function** The scab function displays predicted scab severity level and time until low infection if applicable. Predicted scab severity can be one of four levels: none, low, moderate, or high. Infection time is only displayed if an infection period has started and the scab severity level is none. Infection time is listed as a two digit hour, two digit minute format.	
	Typical Output:	SCAB NONE	Current scab severity level =NONE
		INFT 0810	Time until low scab severity is reached=8 hrs. 10 min.
3		**Sensor Function** The sensor function displays current air and wet bulb temperature, relative humidity, leaf wetness, and rainfall. The current air and wet bulb temperatures are listed in degrees fahrenheit. Their range extends from 0 to 120 F. The relative humidity is computed from the two temperatures and listed as either high (above 90% R.H.) or low (below 90% R.H.). The leaf wetness is described as either wet or dry. The rainfall lists the amount of rain which has fallen in the current infection period in hundreths of inches.	

TABLE VI (Continued)

Typical Output:	TEMP 0073	Current air temp=73 F
	WBTP 0072	Current wet bulb temp=72 F
	RELH HIGH	Current humidity=HIGH
	LFWT WET	Currently the leaves are wet
	RAIN 0002	Two hundreths of an inch of rain have fallen during the current infection period

4 <u>Full History Function</u> (down)

5 <u>Full History Function</u> (up)

The full history function sequentially displays the entire full history buffer which contains stored climatical data about infection periods. This data includes:

a) The date and time of the start of the infection period.
b) The average air temperature during this infection period.
c) The predicted scab severity level.
d) The total period of time during which this infection period occurs.
e) The total time in which the leaves were wet.
f) The total time the leaves were dry and the humidity high.
g) The total time the leaves were dry and the humidity low.
h) The total rainfall during this infection period.

All of these data units (degrees F, hours/minutes) are consistent with earlier definitions.
Upon presenting the data pertaining to the last stored infection period, the function will terminate with the word END of BEGN depending on the direction of travel through the full history buffer. Key 4 sequentially presents data pertaining to earlier infection periods while Key 5 sequentially presents more recent data.

Climatical data pertaining to the infection period of May 22:

DATE MY22	Starting date= May 22
TIME 1212	Starting time=12:12 p.m.
TAVG 66	Average temperature=66 F
SCAB MOD	Predicted scab severity level= moderate
TOTL 1630	Total period time=16 hrs. 30 minutes
WTLF 1400	Total time leaves were wet=14 hrs. 0 minutes
DYLF 0050	Total time leaves were dry and humidity low-50 minutes
HIRH 0140	Total time leaves were dry and humidity high=1 hr. 40 minutes
RAIN 0075	Total rainfall=75 hundreths of an inch

6 Partial History Function

The partial history function sequentially displays the entire partial history buffer which contains stored climatical data about infection periods until low level scab severity occurs. This data includes:

a) The date and time of the start of the infection period.
b) The average air temperatures before low level scab.
c) The amount of time the leaves were wet before low level scab.
d) The amount of time the leaves were dry and humidity low before low level scab.
e) The amount of time the leaves were dry and humidity high before low level scab.
f) The amount of rainfall received before low level scab.

Upon presenting the data pertaining to the last stored infection period, the function will terminate with the word END. The function sequentially presents data pertaining to earlier infection periods.

Climatical data until low level scab pertaining to the infection period of May 22.

DATE MY22	Starting date- May 22

TIME 1212	Starting time-12:12 p.m.
PTAV 0061	Average temperature=61 F
PWET 0800	Time leaves were wet=8 hrs. 0 minutes
PDRY 0030	Time leaves were dry, humidity low= 30 minutes
PHRH 0030	Time leaves were dry, humidity high=30 minutes
RAIN 0010	Rainfall=10 hundredths of an inch

7 Eradicant Spray Function

The eradicant spray function lists fungicidal sprays (eradicants) and the amount of time left in which they can be applied to eliminate scab development on apples. One to five sprays are listed for each infection period where at least one spray is still effective. The format of the output includes date and time of infection period followed by one to five sprays. This format is repeated until an infection period is found where sprays are not effective or no more infection periods exist. The amount of time left for these sprays to be applied is computed twice, once from the beginning of the infection period, and once from the time these infections reach low levels. ERD1 and ERD2 respectively, distinguish these two computations. The word END signifies the termination of ERD1, ERD2 and the function.

Typical Output:

ERD1	Indicates eradicant routine one where spray times are calculated from the beginning of the infection period.
DATE MY22	Date of infection period start.
TIME 1212	Time of infection period start.
CPTN 1241	Captan fungicide and time left to apply it=12 hours 47 minutes.
CYPX 2447	Cyprex "
PHGN 3047	Phygon "

LSUL 5447	Lime Sulfur	"
BLOC 9047	Bloc	"
ERD2	Indicates eradicant routine two where spray times are calculated from the time scab severity becomes low.	
DATE MY07	Date of low infection period.	
TIME 1004	Time of low infection period.	
BLOC 910	Bloc fungicide and time left to apply it=9 hours 10 minutes.	

8 Prediction Stop Function

The prediction stop function provides a means to terminate an infection period via keyboard control. If this function is executed properly, the current infection period will be terminated and the machine reset to accept a new infection period. A proper password must be entered to execute this function. The password can only be entered after XXXX is displayed. If an illegal password is entered, the word ERR will appear in the display and the function will abort. If at any time a user decides not to terminate an infection period after the password is entered, he may do so by not completing the sequence below:

Input	Output	
#	+	
8	PW	Indicates a password is required to proceed with this function.
*	XXXX	Password or Passnumber can be entered at this time.
1234	1234	Passnumber
*	PRED	
*	STOP	Indicates that this is prediction-stop function.
*	+	Prediction has been terminated.

9 Rain Density Function

The rain density function sequentially displays the rain density buffer listing the date and time of the eight most recent heavy rainfall occurrences. Heavy rainfall occurs at any time 3/100 of an inch or more rain falls during a 10 minute interval. After displaying the earliest stored time or reaching the end of the buffer, function 9 displays END.

Typical Output:

DATE JA22	Date of at least 3/100 of an inch of rain in 10 min. on Jan. 22.
TIME 1402	Time of at least 3/100 of an inch of rain in 10 min. at 2:02 p.m.
. . .	
DATE JA08	
TIME 0802 END	

0 <u>Calendar Set Function</u>

Provides a means to set the current date and time via keyboard control. A correct password must be entered before authorization is granted to perform this function. Subsequent to entering a correct password, a correct date and time may be entered immediately following the machine listed date or time. This new date or time must be entered in a purely numerical notation (e.g. NV21=1121, 2:02 p.m.= 1402). All invalid dates and times will cause the message ERR to appear in the display and the function to abort.

Input	Output	
#	+	
0	PW	Indicates a password is required to proceed with this function.
*	XXXX	Password or passnumber can be entered at this time.
5678	5678	
*	DATE	Current date will appear next.
*	JA22	
1121	1121	Enter correct date, number 21.
*	TIME	Current time will appear next.
*	1204	
1402	1402	Enter correct time, 2:02 p.m.
*	+	Function complete.

TABLE VII

CONTROL SWITCHES

Power switch	Connects both internal and extermal power supplies to the circuitry
Reset switch	Resets the apple scab circuitry
Run switch	Starts the apple scab sircuitry
Air temperature Thermistor Trimpot switch	Connects dry thermistor or variable trimpot resistor to the air temperature circuitry
Wet bulb temperature Thermistor Trimpot switch	Connects wet bulb thermistor or variable trimpot resistor to the wet bulb temperature circuitry
Clock switch	Connects fast clock (1 Hz.) or normal clock (1/60 Hz-1 per min.) to the clock input circuitry
Debug switch	Disables or enables the debug mode of operation

circuitry, and debug.

The run and run fast modes deal with starting the apple scab predictor routine at 60 times normal speed. The evaluate temperature circuitry mode allows the entire temperature computing function to be checked. This is accomplished by substituting a calibrated trimpot for the thermistor input to the circuit and then viewing the corresponding output. The debug mode enables the scab predictor to execute debug software. Table VIII summarizes these operational modes.

TABLE VIII

SIGNIFICANT OPERATIONAL MODES

1) Run mode (normal speed)
    a) Turn power switch on (right)
    b) Press reset switch (right)
    c) Press run switch (right)

2) Run fast mode (60 times normal speed)
    a) Follow same procedure as run mode
    b) Place clock switch in 1 Hz position (down)

3) Evaluate temperature circuitry mode
    a) Place both termistor-trimpot switches in thermistor position (right)
    b) Adjust both temperature trimpots to 10K ohms
    c) Place both termistor-trimpot switches in trimpot position (left)
    d) Place clock in the 1 Hz position (down)
    e) In 20 seconds read both dry and wet bulb termperatures off the LCD (temperature=87F)

4) Debug mode
    a) Enable debug mode by activating debug switch (up)
    b) Connect and activate microterminal

The debug section allows the operator to debug both hardware and software on the machine level. This feature is made possible by the connection of an RCA evaluation kit processor board and a microterminal. The processor board contains a series of LED's which display the internal busses and states of the predictor machine (memory bus, data bus, control and state lines). The microterminal contains a 24-button keyboard and 8-digit numeric display which can be used to control and monitor the machine.[4]

Each key of the microterminal's keyboard performs a specific operation. These keys and operations are listed in Table IX. These operations in many aspects are similar to those described in the evaluation kit.

The software routines and typical examples of the usage of debug hardware/software are listed below:

CASE I Memory Inspect Modify

Memory inspect/modify allows the operator to inspect all memory locations and modify RAM locations. The memory address is listed on the left followed by the memory data on the right of the microterminal display.

Change RAM location 1001 to have data 35:

PRESS	DISPLAY		
RU	0 0 0 0	7.1.	Memory data
1001	1 0 0 1	2.2.	
⟷	1.0.0.1.	2 2	
35	1.0.0.1.	3 5	
INC	1.0.0.2.	6 1	

TABLE IX

MICROTERMINAL CONTROL/DEBUG FUNCTIONS

KEY	ACTIVITY
R (Reset)	The reset key places the scab predictor in the reset mode.
RP (Run Program)	The run program key places the scab predictor in the run mode.
Step/Cont (Step/Continous)	If the step/continous key is set to step the microprocessor will be forced to operate in a single cycle mode. This mode allows the operator to inspect each cycle the machine performs via the attached evaluation key processor board's LEDs.
RU (Run utility software)	The RU switch stops the execution of the scab predictor software and starts the execution of the debug software in the predictor machine. This is only active when the machine is set for debug mode. (debug switch up)
CA (Change routines)	The change routines key allows the operator to select one of the following five debug routines: a) Memory inspect/modify b) Internal register inspect/modify (1802-RO-RF) c) Internal register inspect/modify (1802-DF-XP-D) d) Hardware output (4514) e) Hardware output (4508) Each depression of the change routine key terminates the current routine and activates the next in a cyclic manner.
↔ (Change decimal position)	The ↔ key changes the decimals on the microterminal display from the right side to the left or vice versa depending upon their initial position. The decimals protect the portion of display they are covering from modification.
INC (Store/increment to next location/output)	The INC key performs various functions depending upon which debug routine is being activated. During inspect/modify routines INC will store the inspected data, increment to the next memory or register location, and display the respective data. During hardware output routines INC outputs the specified information.
$P (Return to scab program)	$P terminates debug software execution and returns control of the machine to the predictor software.

CASE II Internal Register Inspect Modify (RO-R15)

Internal register inspect/modify (RO-R15) allows the operator to inspect and modify all 16 bit internal registers of the 1802. The register number is listed on the left followed by the register data on the right of the micro-terminal display.

Change R3 to 07AA:

PRESS	DISPLAY		
	1.0.0.2.	6 1	
CA	0	0.5.B.C.	Register data
3	3	1.0.0.0.	
⟷	3.	1 0 0 0	
07AA	3.	0 7 A A	
INC	4.	0 8 B A	

CASE III Internal Register Inspect Modify (DF-XP-D)

Internal register inspect/modify (DF-XP-D) allows the operator to inspect and modify the DF, XP, and D registers of the 1802 microprocessor. The display format consists of DD, DF, XP and D where DD is the routine identifier, DF is 1 or 0 in the 4th display location, XP occupy the 5,6 display locations, and D occupies the 7,8 display locations.

Change D-FF:

PRESS	DISPLAY	
	4.	0 8 B A
CA	d d	1.2.3.5.7.
⟷	d.d.	1 2 3 5 7
123FF	d.d.	1 2 3 F F
INC	d.d.	1 2 3 F F

CASE IV Hardware Output

Hardware output 4514/4508 allows the operator to output a hex digit to the 4514/4508 chips. The output devide is listed on the left followed by the selected hex digit.

Output E to 4514:

Output C to 4508:

PRESS	DISPLAY	
CA	4 5 1 4	0.
⟷	4.5.1.4.	0
E	4.5.1.4.	E
INC	4.5.1.4.	E
CA	4 5 0 8	0.
⟷	4.5.0.8.	0
C	4.5.0.8.	C
INC	4.5.0.8.	C

CHAPTER III

HARDWARE ORGANIZATION

This chapter describes the hardware organization of the ASDP. Figure 3.1 illustrates the principal ingredients which comprise the hardware. For simplicity, the ASDP can be divided into four functionally distinct hardware sections: control, input, output, and support. The control section as described in Section 3.1, is the cerebrum and cerebellum of the ASDP. Its functions include the evaluation and storage of data along with the coordination of all instrument operations. Likewise, the input section, Section 3.2, and output section, Section 3.3, form the environmental interface of the ASDP. Along with the control section, they function to complete the data transfer loop from sensors to user by converting the raw data into a machine format on one hand and converting the machine format to user format on the other. Finally, the support section, described in Section 3.4, forms the backbone of the ASDP. Its function is to provide an environment in which the other three sections can operate efficiently.

3.1 Control Section

The control section contains three subdivisions: processor, memory (ROM and RAM), and control circuit (Figure 3.1). These operate in unison to coordinate the sequencing of the instrument.

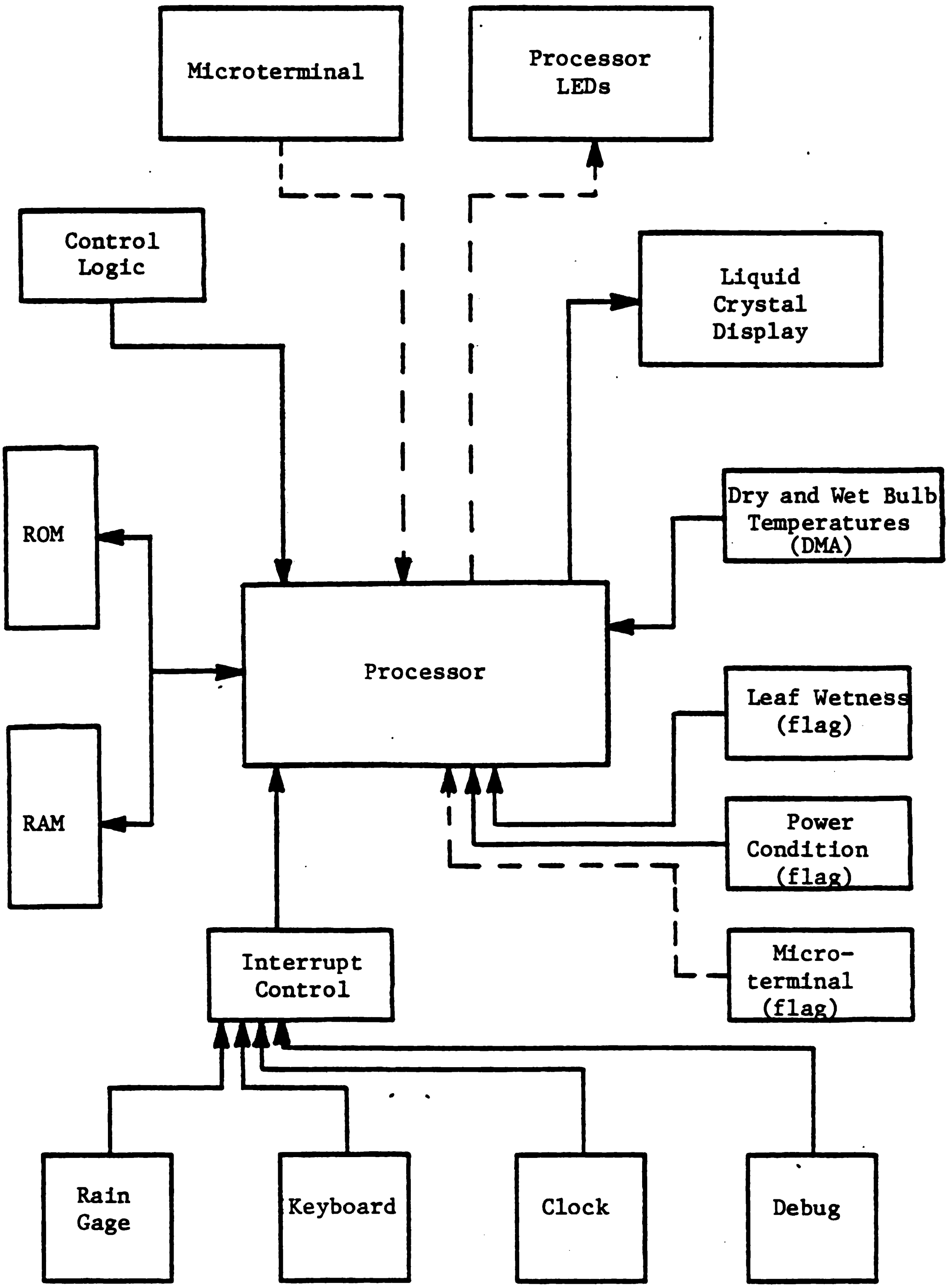


Figure 3.1 -- Block Diagram of ASDP

3.1.1 Processor

The processor section dutifully performs the function of sequencing the hardware and implementing the ASDP program. The realization of this performance is achieved by a microprocessor and two output decoding chips. A microprocessor was selected as the processor because it can easily be programmed to the level of sophistication required by the ASDP (This sophistication level requires both algebraic and logical manipulation of numbers in the range of 0 to 64000, and complex decision analysis based upon various inputs and the results of algebraic solutions). Furthermore, a microprocessor is clearly the only choice for the processor function when the cost, power consumption, and interface requirements of other processors are considered. These same considerations (power consumption, interface requirements) were also taken into account in the selection of the specific microprocessor to control the ASDP--the RCA 1802 COSMAC Microprocessor. The 1802 is ideally suited for this instrument because of its hardware and software features. Three of the most significant include: single voltage supply, TTL compatability, and extremely low power requirements. Additional features of the 1802 are listed in Table X and described in the 1802 user's manual.[5] Finally, the ease of using the 1802 is enhanced by the availability of numerous support chips,[6] a cross-assembler simulator,[7] and COSMAC Evaluation Kits.[8]

An N-bit (3-bit) 1-of-8 decoder (1853) and 4-bit 1-of-16 decoder (4514) complete the processor section hardware by interfacing with the 1802 to provide input and output timing pulses. These timing pulses are required by many of the ASDP circuits to activate functions, to reset conditions, and to synchronize data transfer with the

TABLE X

ADDITIONAL FEATURES OF THE 1802

Internal

8-bit processor

92 instructions
- Addition, subtraction, logical
- Branch
- Store and load
- Control
- Register-manipulate

16-bit memory addressing (64K)

16 internal registers to act as
- Program counters
- Data pointers
- Data registers

Input

Four testable flags

Two direct memory access (DMA) lines

One interrupt line

8-bit data bus with N-bit code

Output

One Q-programmable data register

8-bit data bus with N-bit code

Memory Control

Memory read line	MRD
Memory write line	MWR

Control Lines

Wait control	WAIT
Clear/reset control	CLEAR

Timing Pulses

Timing pulse A	TPA
Timing pulse B	TPB

microprocessor. Specifically, the 1853 activates one of seven lines upon reception of a valid N-bit code (input or output instruction) from the 1802 (See Figure 3.2). This activated line then selects or enables one of the ASDP circuits so that data transfer can occur between this circuit and the 1802. The 4514, on the other hand, uses 4-bits of the data bus along with 2 of the decoded lines from the 1853 to perform a 1-of-16 decoding operation. This allows 16 additional output timing pulses to be defined. These output pulses synchronize display data transfer, reset interrupts, and activate the temperature circuits. Total operation of the 4514 is completely controlled by the 1802 by way of the 1853. Typical operation of the 4514 involves, first, the latching of the data word which is to be decoded. This operation is performed when the 1802 executes an OUT 2 or 62 instruction. With this instruction a 4-bit decode data word is placed upon the data bus simultaneously with the proper N-bit sequence. The N-bit decoder then activates line 2, which in turn, latches the decode data word into the 1-of-16 decoder. A second output by the 1802 then enables the 1-of-16 decoder to utilize the latched decode data word to force the selected line high. This action occurs upon the OUT 1 or 61 instruction execution. During this time (61 execution), the 8-bit data bus can present 8 bits of data for the selected ASDP circuit if it requires it. Therefore, with the 4514 at least, 16 8-bit data words can be output. Refer to the example shown along with Figure 3.2 for complete operation of this circuit.

3.1.2 Memory

The function of the memory section is to store the scab prediction program and retain prediction data. High density, non-volatile EPROM

Example:

Machine Code	
	SEX RO
	OUT 2
	.01
	OUT 1
	.X'FF'

Data lines	SEX RO	OUT 2	OUT 1
NO	0	0	1
N1	0	1	0
N2	0	0	0
OUT 2	0	1	0
OUT 2	0	0	1
Data Bus	X'00'	X'01'	X'FF'
SO-S15	All 0	All 0	S1 = 1 All others = 0

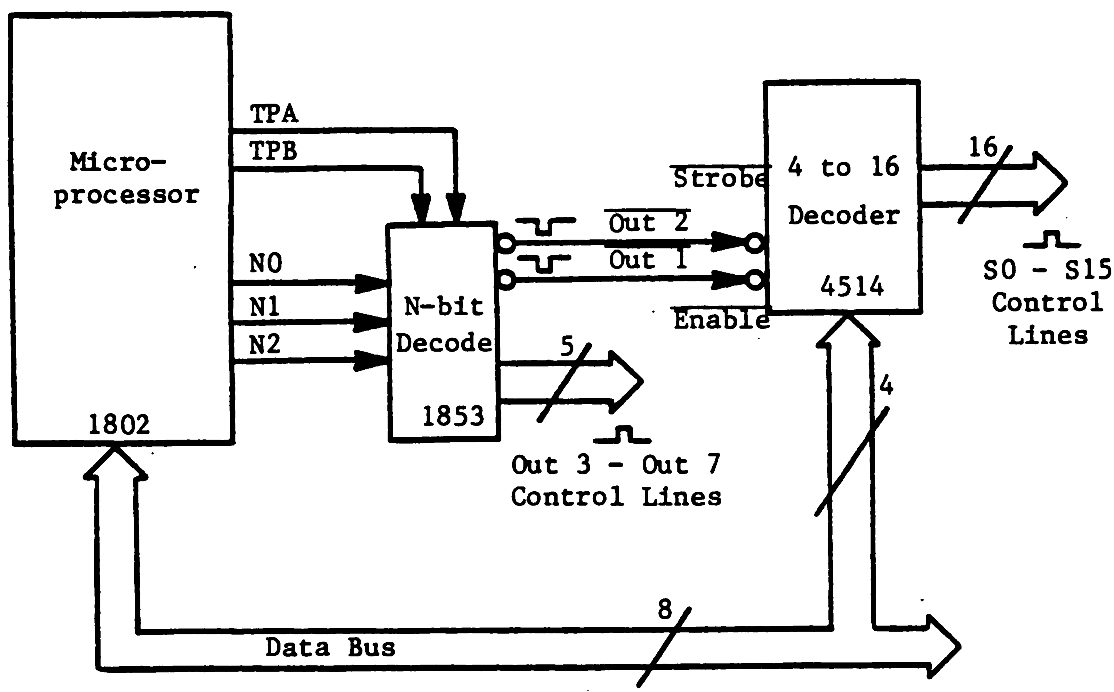


Figure 3.2 -- Processor Section and Example

(electrically programmable UV-erasable read only memory) and low power CMOS RAM (random access memory) in conjunction with a decoding circuit perform this function (Figure 3.3). The total memory space when two EPROM chips (Intel 2716, 8-bit x 2048 word) and four RAM chips (RCA 1822 4-bit x 256 word) are coupled is 4608 8-bit words. The 4096 ROM words are dedicated to program storage, while the 512 RAM words are almost exclusively used for prediction data storage. Specifically, the RAM stores all current weather, prediction, display and buffer (described in Section 2.3) data. In addition, two of the RAM locations store program instructions to execute a continuous loop between interrupts. This loop reduces the power consumed by the entire ASDP by 60 percent by placing the EPROMs in a power down mode. The decoding circuit segments the memory into four blocks as listed in Table XI. These blocks are defined by a memory addressing scheme which is compatible with the memory size. Two chips, 1859 and 4076 implement this decoding scheme by keying on selected address bits to produce chip enable pulses. These selected address bits are listed in Table XII. The block of memory which is enabled is then allowed to utilize the data bus for data input and output.

3.1.3 Control Circuit

The function of the control circuit is to initiate program execution and facilitate hardware and software debugging. Three switches perform this function by placing the machine into one of the three modes listed in Table XIII. Proper initialization of the program involves depressing the circuit board mounted momentary reset switch followed by the momentary run switch. Depression of the reset switch as

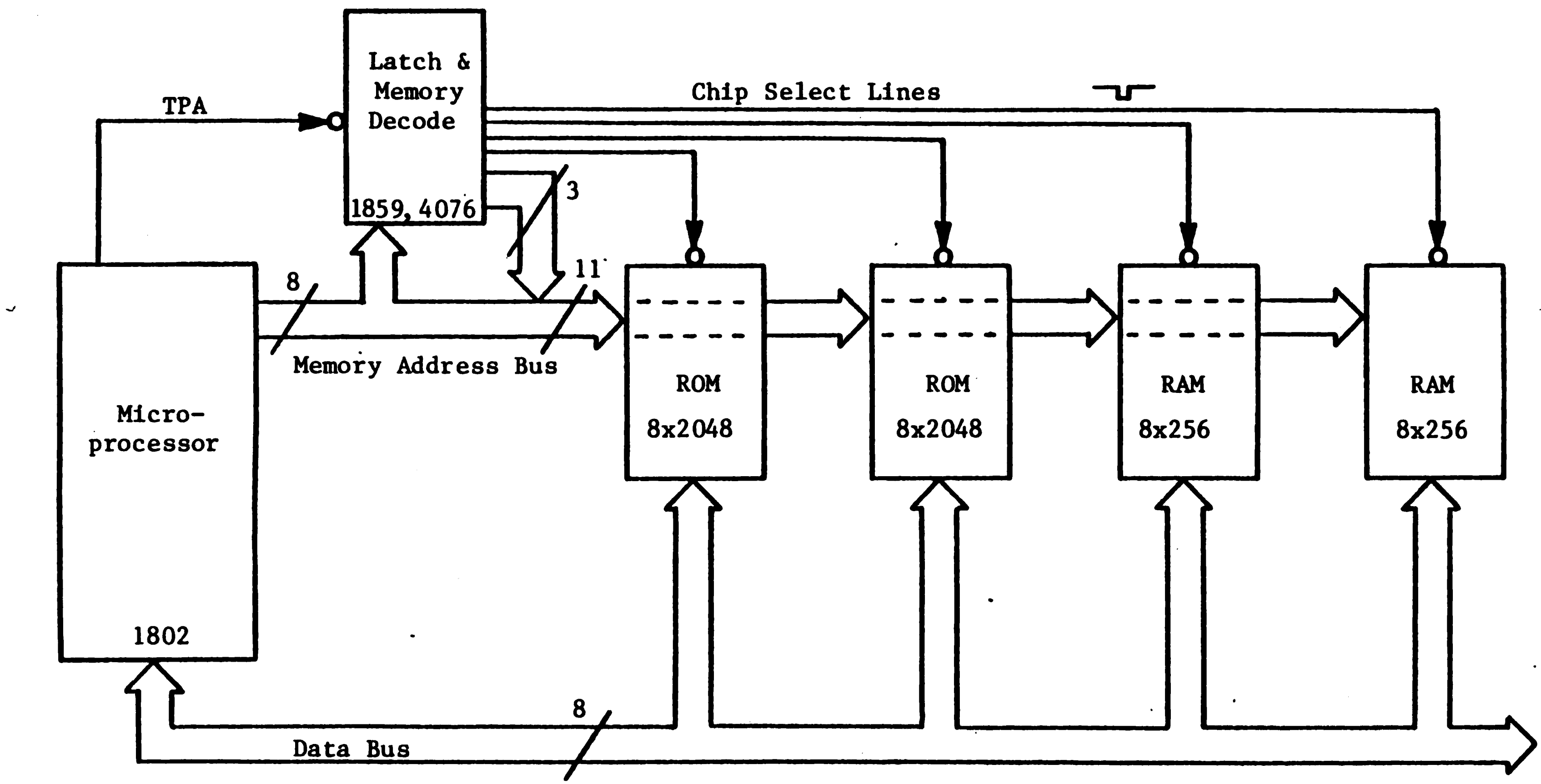


Figure 3.3 -- Block Diagram of Memory Circuit

TABLE XI

MEMORY BLOCKS

Memory Type	Purpose	Address	Part #	Space
Read only	Program store	0000-07FF	2716-1	2K
Read only	Program store	0800-0FFF	2716-2	2K
Read/write	Data store	1000-10FF	1824-1-2	256 words
Read/write	Data store	1100-11FF	1824-3-4	256 words

TABLE XII

SELECTED ADDRESS BITS FOR DECODING

Bit 12	Bit 11	Bit 8	Address Range	Part Enable
0	0	X	0-2047	2716-1
0	1	X	2048-4095	2716-2
1	0	0	4096-4351	1824-1-2
1	0	1	4352-4607	1824-3-4

Note: The X's indicate "don't cares".

TABLE XIII

MACHINE MODES

Switch	Mode
Reset	Reset
Run	Run
Step	Single Cycle

shown in Figure 3.4 sets the clear flip flop which in turn initializes the microprocessor and interrupt circuitry. The reset mode is entered when this initialization process is complete and the reset switch is released. Depression of the run switch simply resets the clear flip flop and places the machine in run mode. Upon entering the run mode, the predictor program should begin to execute.

Program execution is halted by placing the step switch upon the RCA Microterminal into the single cycle position, thus entering into the single cycle mode. Closing the step switch allows the wait control line to go low during the timing window produced by the two timing pulses, TPA and TPB. Once the wait line goes low, the microprocessor will stop execution upon the next falling edge transition of the two MHz clock oscillator. Execution of the program will not continue until either the step switch is opened or the run switch is depressed in which case only one machine cycle will be performed.

Each depression of the run switch produces a rising edge similar to the TPB timing pulse which opens the window function or places the wait line high. The wait signal remains high and allows one cycle to execute until the falling edge of TPA at which time it drops low and requires another run pulse to continue execution.

In addition, if the COSMAC Evaluation Kit is connected to the predictor circuitry the single cycle function can become an effective debugging aid. The LED's on the processor board allow the operator to monitor the CPU address bus, data bus, and control lines between each cycle of execution. This information can then be utilized to determine if the hardware and software are functioning properly.

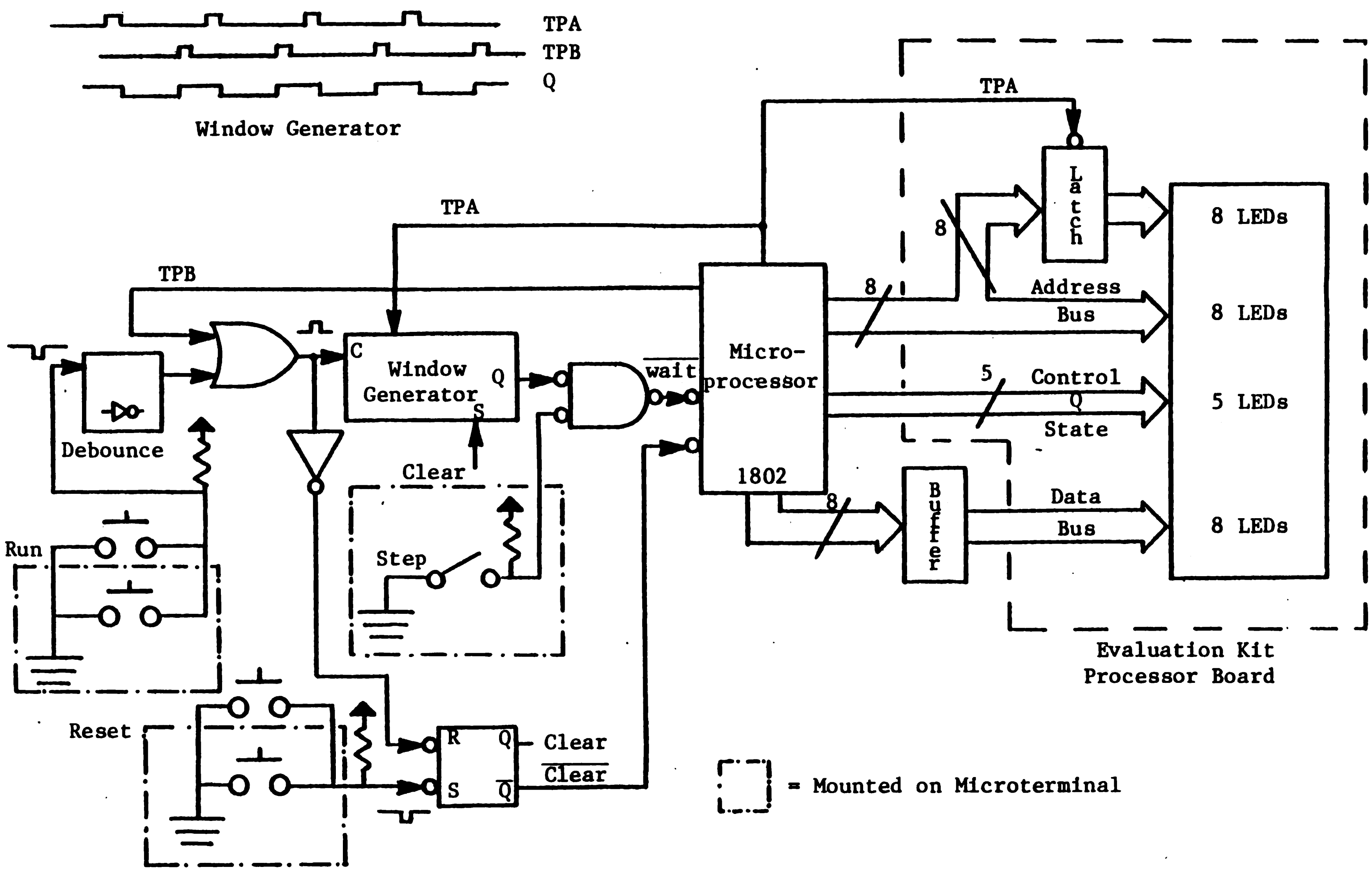


Figure 3.4 -- Control and Debug Circuit

3.2 Input Circuits

The input circuits utilize three mechanisms--interrupts, flags, and DMA (Direct Memory Access)--to input data. These mechanisms, illustrated in Figure 3.1, allow events as well as conditions to be input to the ASDP.

3.2.1 Input Via Interrupts

Interrupts provide the means for the predictor machine to record and respond to randomly occurring events. Figure 3.1 illustrates the total interrupt organization. As shown in the figure, four random events (See Table XIV for descriptions) must be monitored. Each occurrence of an event produces a pulse by way of a specialized event circuit. Direct monitoring of these random pulses is then performed by an interrupt control circuit. Decisions are then made by this circuit as to what action the microprocessor should take. This action then dictates the apple scab predictor's response to these events.

The interrupt control circuit is designed around the constraints listed in Table XV. The first five of these constraints (enumerated in Table XV) were designed into the hardware as shown in Figure 3.5. The clocked D-type flip flops provide edge triggering as well as interrupt request storage. The processor-controlled mask then determines which stored interrupt requests are enabled. The enabled interrupts are then latched to provide a stable 4-bit interrupt status word to the mircoprocessor. The microprocessor then reads the status word (after the enabled interrupts pull the interrupt line low) by enabling the latch to place it on the data bus. The status word is then decoded to determine interrupt priority. Table XVI lists the priority levels associated

TABLE XIV

DESCRIPTION OF RANDOM INTERRUPT EVENTS

Event	Description
Rainfall occurrence	1/100 of an inch of rain has fallen
Keyboard service request	User is operating keyboard
Clock occurrence	One second or minute has passed
Debug service request	Operator wishes to utilize debug routines

TABLE XV

DESIGN CONSTRAINTS FOR INTERRUPT CONTROL CIRCUIT

1. All interrupt requests or events are to be recorded upon an edge transition so that requests of varying lengths are only acknowledged once.
2. All interrupt requests are to be stored until the machine is available to respond to them.
3. An individual reset of each recorded interrupt request is to be microprocessor controlled.
4. A mask is to be used to disable specific interrupts at critical transitions in the program.
5. The interrupt status is to appear stable and defined during the time the microprocessor is attempting to read it.
6. An interrupt priority system is to be defined.

TABLE XVI

INTERRUPT PRIORITY LEVELS

Priority Level	Interrupt
Highest	Rainfall
Second	Keyboard
Third	Clock
Lowest	Debug

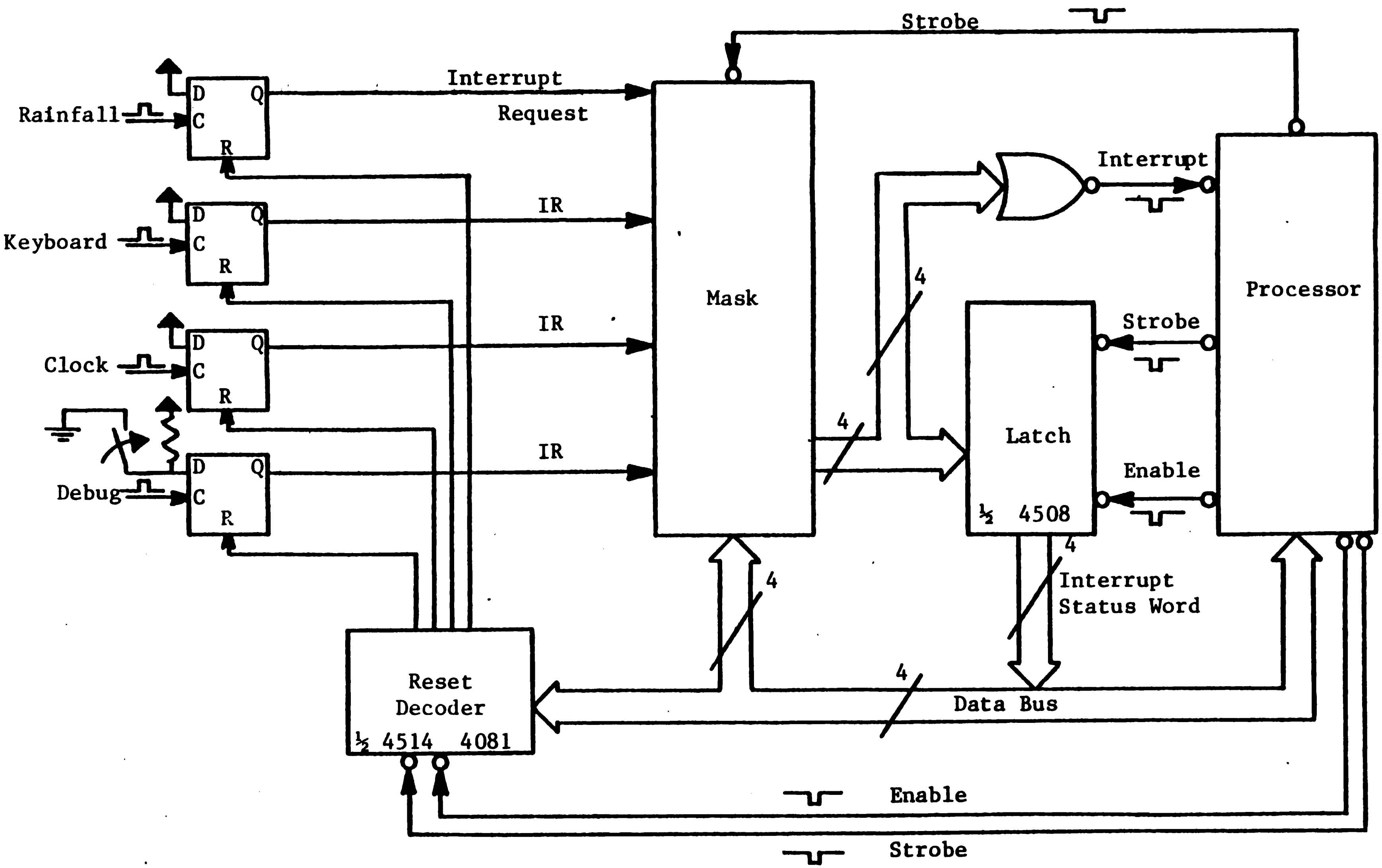


Figure 3.5 -- Interrupt Control Block Diagram

with each of the event interrupts. The priority level determines which interrupt event is to be responded to first in cases of multiple interrupts. The highest priority pending interrupt is then executed via its specific routine (rain, keyboard, clock, debug).

Upon the completion of the interrupt routine, the microprocessor resets the appropriate interrupt request storage D-type flip flop, enables the appropriate interrupts (masks), and returns to its last place of execution. At this point the execution of a single interrupt is complete and the interrupt control circuit is poised to process another one of the four interrupts produced by the specialized event circuits (rain, keyboard, clock, and debug).

3.2.1.1 Rain

The rainfall sensor provides the predictor with information about the intensity of and amount of rainfall that has occurred. This data is presented in the form of a pulse upon each accumulation of 1/100 of an inch of rain. A tipping bucket rain gage built by Weathertronics collects the rainfall in a miniature bucket and momentarily closes a switch after the bucket has filled.[9] The closing of the switch produces a 100 ms pulse which is debounced by the software.

3.2.1.2 Keyboard

The keyboard and associated circuitry form the input portion of the user-machine communication link. Activation of the weather-resistant elastomer-action keyboard produces an interrupt request by way of the keyboard encoder chip. The encoder chip submits this request only after all of the necessary debounce, two-key rollover, key encoding, and storing operations are performed. The stored last key depression data is then passed to the microprocessor via a request through the keyboard interrupt routine. Software decodes which key was depressed and determines subsequent action.

3.2.1.3 Clock

The precision clock oscillator, coupled with a 4,194,304 Hz crystal, synchronizes the scab prediction operation by requesting interrupt service at precisely timed intervals. These intervals are of one second or minute time durations, depending upon the position of the clock switch mounted upon the circuit board (refer to Figure 3.6).

3.2.1.4 Debug Interrupt

The debug interrupt circuitry formats the debug interrupt which, in turn, informs the microprocessor that debug software is to be executed. The three major components which form this circuitry as shown in Figure 3.6 include the microterminal, the debug switch, and a TPA pulsed counter. The switch connects directly to the debug interrupt request flip flop and enables or disables its action. Once debug execution is requested, the microterminal provides the display and keyboard, which the debug software makes use of. Debug breakpoints can then be executed via the counter chip.

A debug interrupt can be requested in either of the following two ways: user depresses RU on the microterminal, or the microprocessor sets the Q flip flop.

The purpose of setting the Q flip flop is to initiate an interrupt request and allow the operator a means to execute breakpoints through

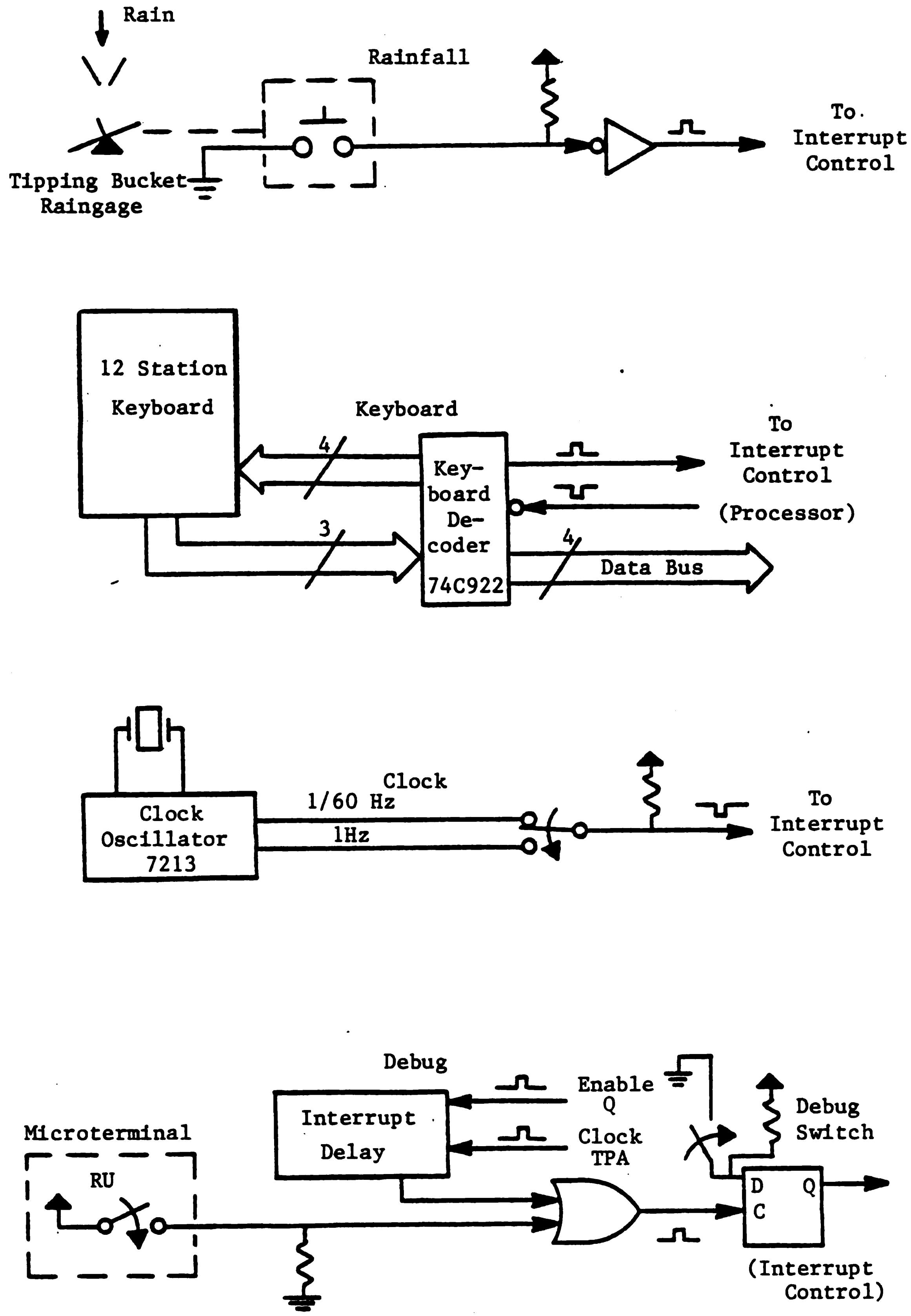


Figure 3.6 -- Specialized Event Circuits

debug. Although this feature was never utilized, it could easily be implemented as outlined below.

1. Press RU to commend debug mode.
2. Enter breakpoint address into the microterminal.
3. Press $P to return to the main predictor program.

The debug program notes the fact that breakpoints have been entered. Therefore, it sets the Q flip flop one instruction before it returns to the main predictor program. With Q being set, only one instruction of the main program will execute before the debug interrupt occurs. Upon entering the debug software again, the debug program notes that the Q is set, and realizes that a breakpoint check must be made. If the current location of the main program doesn't match with any of the breakpoint addresses, the process must be repeated. Otherwise, the process is discontinued because a breakpoint has been reached.

3.2.2 Input via Flags

Flags are lines that the microprocessor can test for a true or false condition, and then respond to accordingly. The three inputs which utilize flags are the leaf wetness sensor, the power condition sensor, and the debug microterminal.

3.2.2.1 Leaf Wetness

The leaf wetness sensor circuitry monitors the degree of moisture found upon apple tree leaves. The monitoring function is performed by a leaf wetness sensor and an interpreting circuit. The sensor, a plastic tube on which two non-connecting wires spaced .1 inch apart are wound, presents the interpreting circuit with a resistance which corresponds to the degree of wetness. A typical plot of this resistance vs. the amount of moisture upon the leaves is illustrated in Figure 3.7. The interpreting circuit, two voltage comparators, and a flip flop (Figure 3.8), condition this resistance to form a digital two-state output. The first stage of this conditioning involves converting the input sensor resistance into a voltage which corresponds to the following equation:

$$V_{in} = 5 \frac{220 \times 10^3}{220 \times 10^3 + R_{sensor}} \text{ Volts}$$

This voltage is then compared via the comparators with low and high voltage reference levels to determine the signals which are to be submitted to the flip flop. The flip flop then specifies the state of the leaves as being either wet or dry by controlling one of the microprocessor's flag lines. The two comparators, in combination with the flip flop incorporate a large amount of hysteresis into the wetness detection as illustrated by Figure 3.9 and the associated example. This hysteresis is a desirable function since it accurately defines the type of wetting periods that are to be recorded.

3.2.2.2 Power Condition

The power condition circuitry monitors the main voltage supply to sense power outages and reductions. A voltage comparator as illustrated in Figure 3.10 performs this operation by continually comparing the main supply voltage with a reference voltage. Each reduction or disconnection of power triggers the power flag which, in turn, directs the microprocessor to display the power message warning (POWR). This message indicates to the user that the main power supply is non-operable.

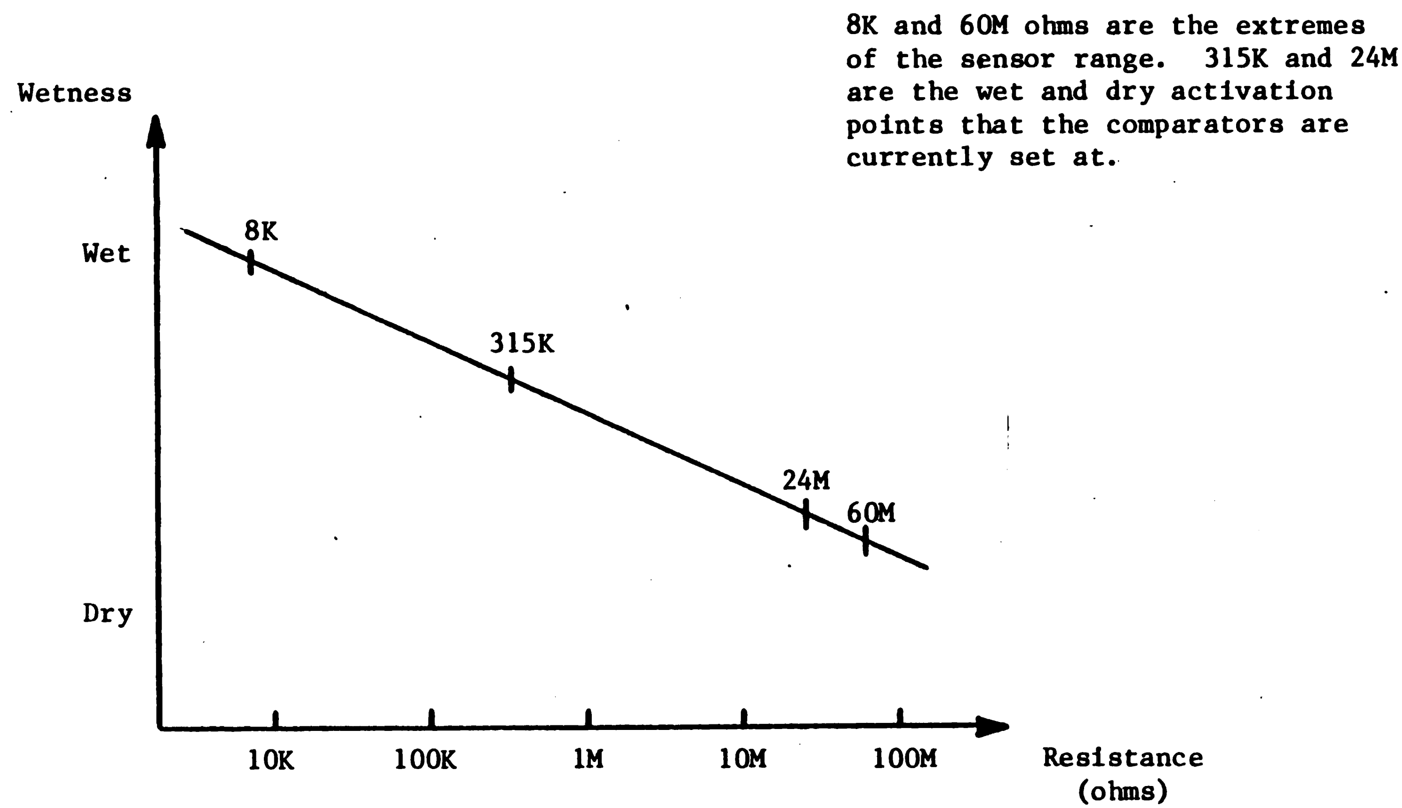


Figure 3.7 -- Sensor Wetness vs. Sensor Resistance

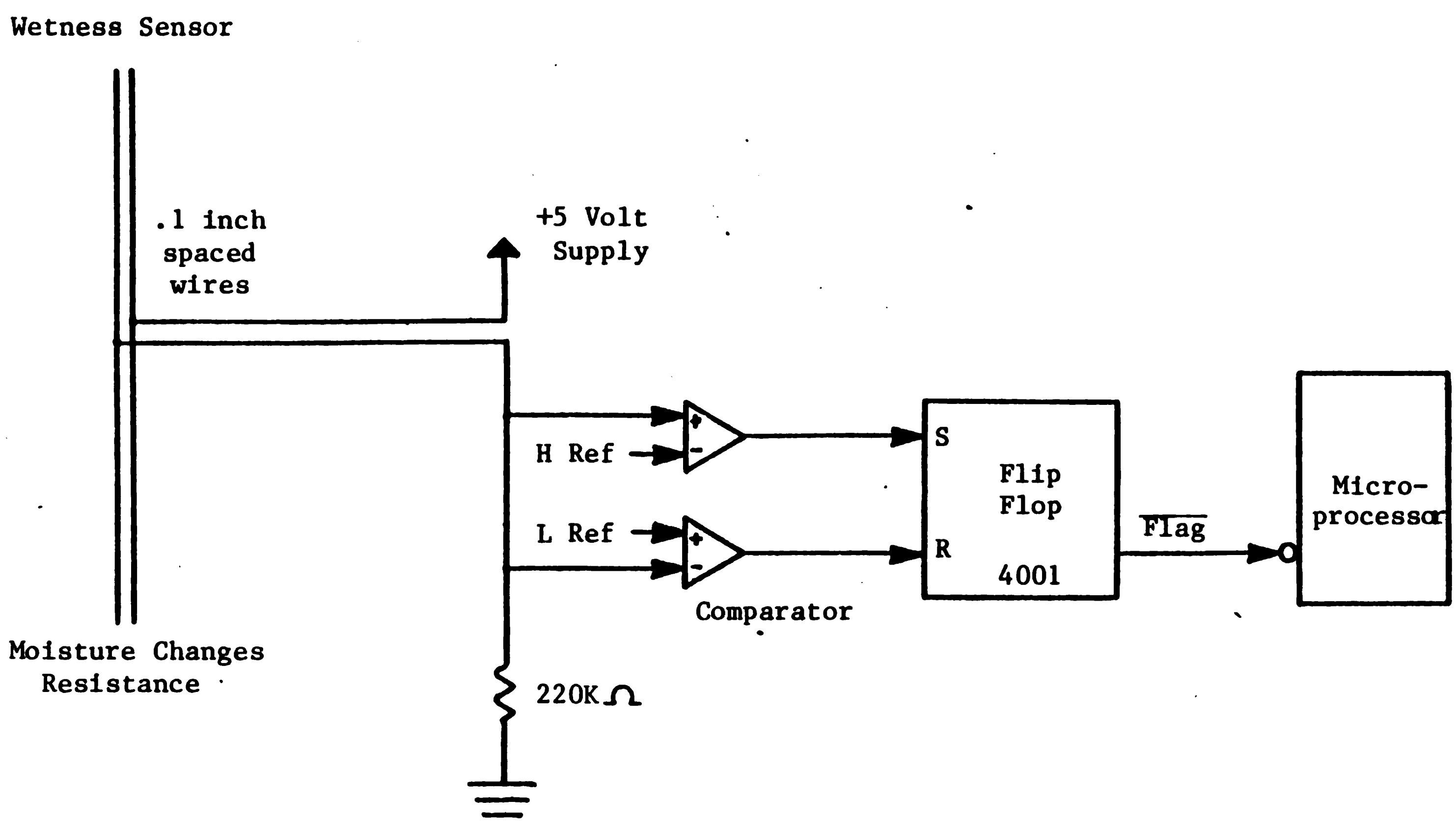


Figure 3.8 -- Leaf Wetness Interpreting Circuit

Example Statements

a) Refer to Figure 3.8 for circuit block diagram.

b) V_{in} represents sensor voltage as determined by

$$V_{in} = 5 \frac{220 \times 10^3}{220 \times 10^3 + R_{sensor}} \text{ Volts}$$

c) L and H represent comparators reference voltages of .05 and 2.0 volts.

d) L and H voltages correspond to 24M and 315K ohm resistances respectively.

Actual Leaf Condition	Comparators Inputs	Comparators Outputs	Flip Flop Output	Machine Interpretation
dry	$V_{in} < L$	high	high	dry
	$V_{in} < H$	low		
damp	$V_{in} > L$	low	high	dry
	$V_{in} < H$	low		
wet	$V_{in} > L$	low	low	wet
	$V_{in} > H$	high		
damp	$V_{in} > L$	low	low	wet
	$V_{in} < H$	low		
dry	$V_{in} < L$	high	high	dry
	$V_{in} < H$	low		

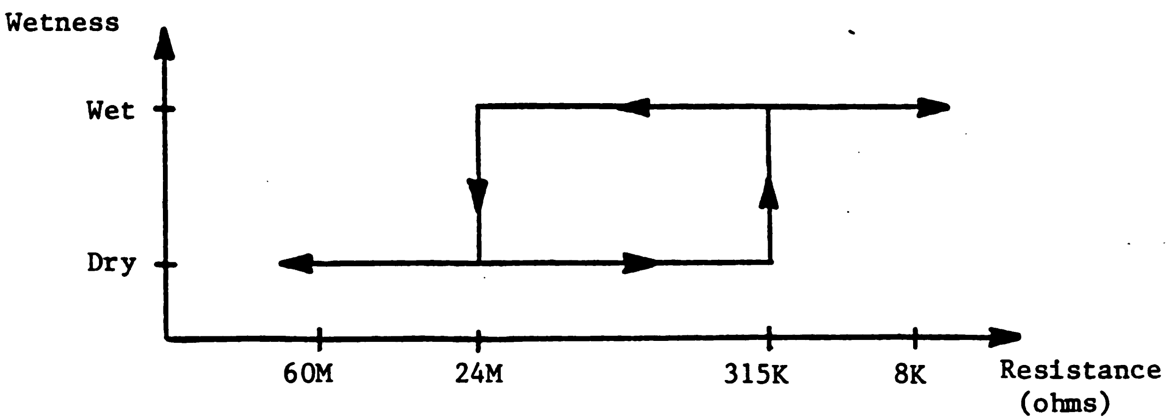


Figure 3.9 -- Hysteresis of Leaf Wetness Circuit

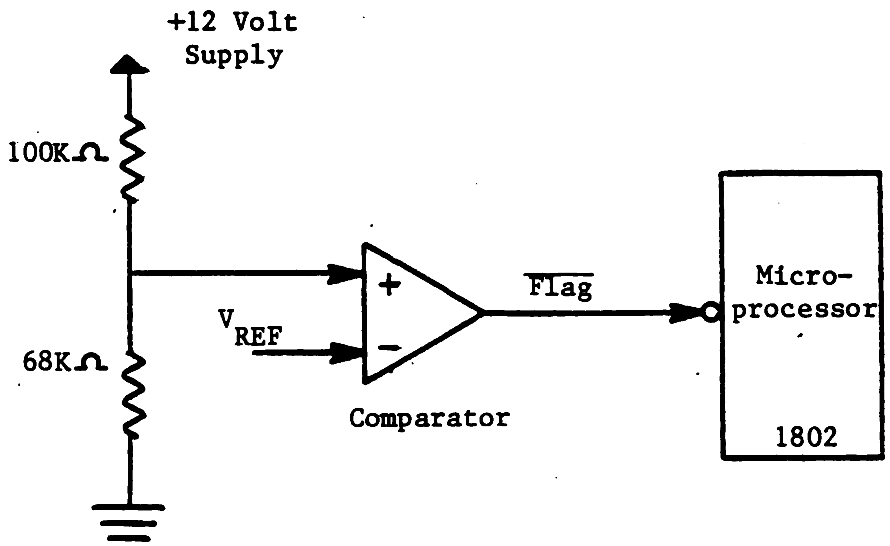


Figure 3.10 -- Power Condition Circuit

3.2.2.3 Debug Microterminal

The microterminal flag completes the data communication link between the microterminal and microprocessor. Once data is entered into the microterminal, its flag signals the microprocessor that data is available. This signaling only evokes a response if debug software is being executed.

3.2.3 Input via DMA

The Direct Memory Access (DMA) mechanism is utilized to measure temperatures. The current dry and wet-bulb air temperatures are obtained by utilizing microprocessor software to equate a given temperature to a regulated pulse width. Figure 3.11 illustrates this procedure. The circuit components that perform this measurement function include two precision termistors, a monostable, a flip flop, transmission gates, and the microprocessor. The precision thermistors (FENWAL #LTN-2), which are the keys to the entire circuit (Figure 3.13), change resistance linearly with temperature as illustrated in Figure 3.12. Therefore, one of the termistors in conjunction with a temperature-stable capacitance form a linear varying time constant. This time constant is then incorporated within a monostable to produce a linear temperature-regulated pulse width. This temperature-regulated pulse width then activates the microprocessor DMA out line by forcing it low until the pulse is complete. This action causes the microprocessor to execute successive DMA out.

DMA out involves outputting the contents of memory specified by R(0) followed by the incrementing of R(0). Therefore, the duration of the temperature-regulated pulse determines the number of DMA cycles

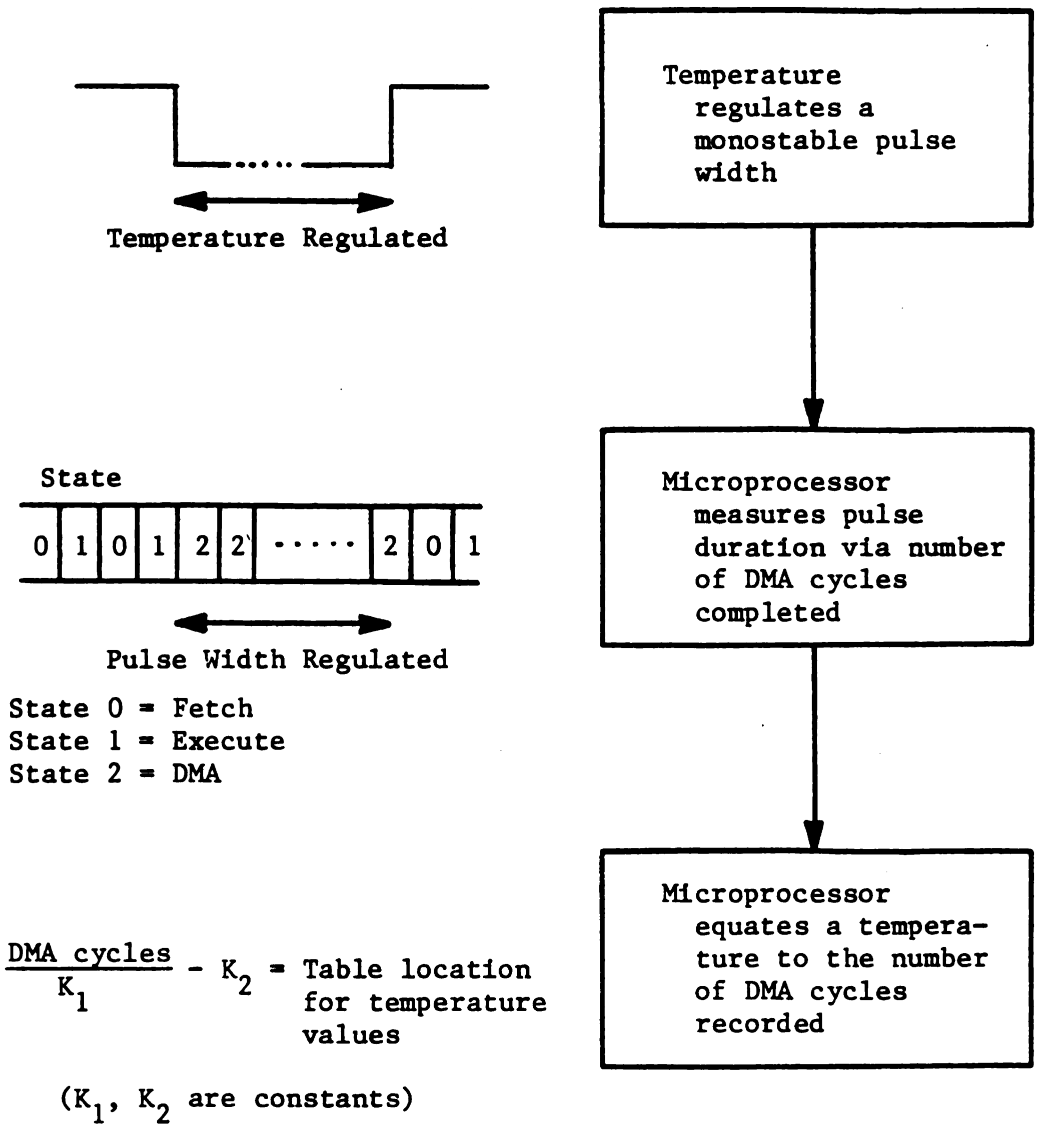


Figure 3.11 -- Procedure for Temperature Measurement

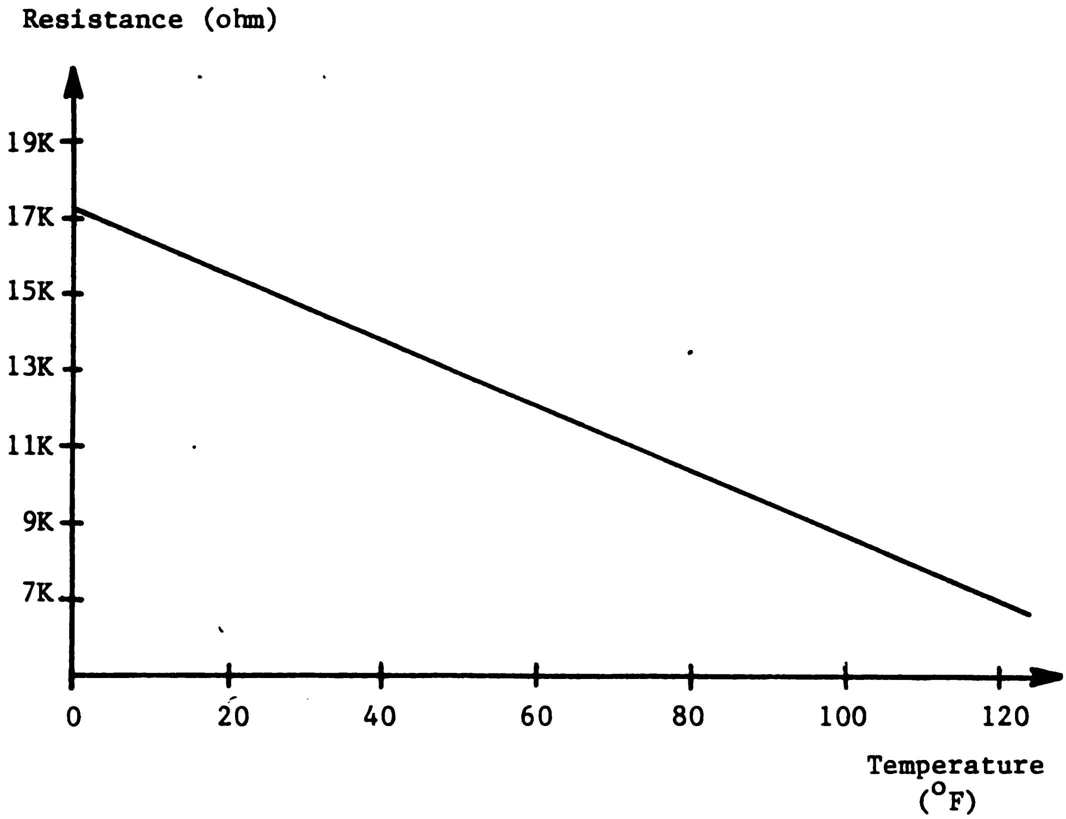


Figure 3.12 -- Percision Thermistor Resistance vs. Temperature

executed or the count in R(0). This count is then converted into the temperature by way of division and a look-up table. Typical R(0) counts per degree are listed in Table XVII.

The calculation of the temperature from this R(0) count will be explained in the next chapter. The remainder of this section will discuss various attributes and problems of the circuit. One feature of the circuit is that it is fully microprocessor-controlled. The microprocessor selects which temperature (dry bulb, wet bulb) is to be measured by choosing the appropriate thermistor via a selection circuit. The selection circuit, composed of a flip flop and transmission gates places the thermistor in the circuit with the monostable and capacitor as shown in Figure 3.13. Triggering of the monostable is then performed by the microprocessor when it is ready. A second feature of the circuit is that the temperature measurement process is exceedingly short. The duration of the temperature-regulated pulse depends on temperature and can last anywhere from 3.1 ms at 120 F to 6.8 ms at 0 F. These calculations are based on the fact that each DMA cycle requires 3.8147 microseconds and that there are 812 and 1780 cycles at each of the respective degrees. One problem with the circuit is that linear temperature variation yields slightly non-linear pulse width variation as shown in Figure 3.14. This is due to the non-linear effects of the circuitry components. In particular, the transmission gates and capacitor together increase the pulse width by an amount equivalent to a 4°F error at low temperatures. On the other hand, the monostable will decrease the pulse width by an amount equivalent to a 1°F error at low temperatures. This leads to a three degree net decrease in temperature (Figure 3.14) or non-linear effect which must be compensated for through software.

TABLE XVII

R(0) COUNTS PER DEGREE

Temperature	R(0) Count (DMA cycles completed)
0	1820
10	1738
20	1656
30	1573
40	1489
50	1407
60	1324
70	1242
80	1161
90	1080
100	1000
110	920
120	839

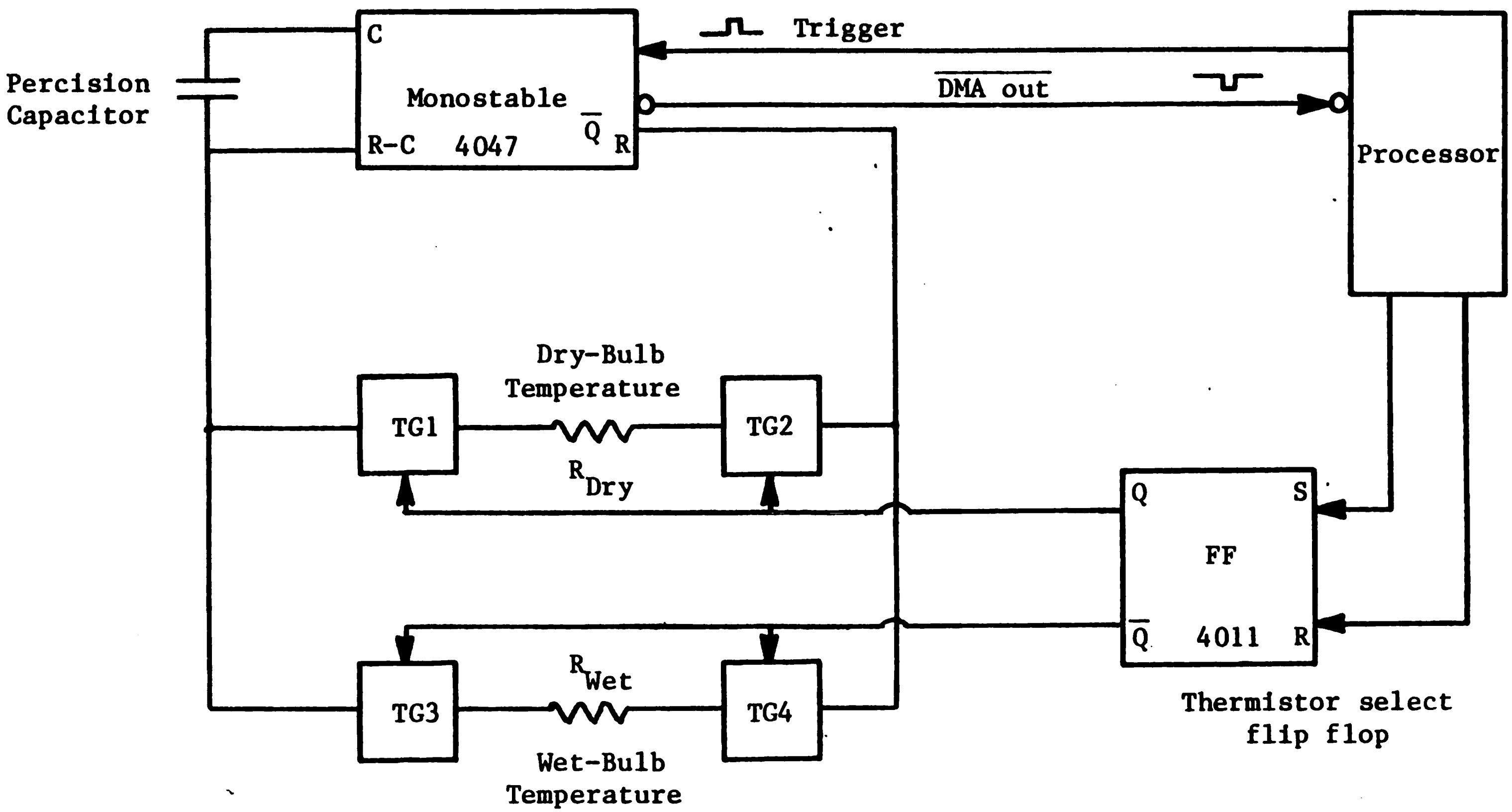


Figure 3.13 -- Temperature Measurement Circuit

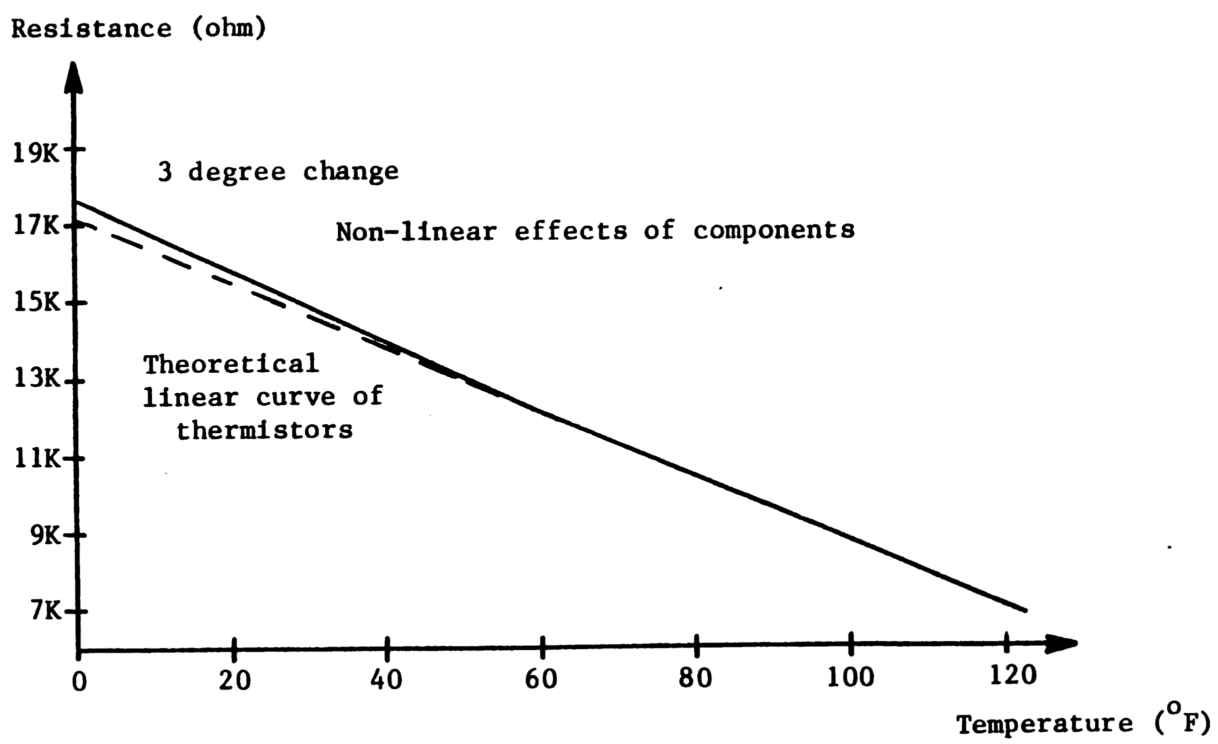


Figure 3.14 -- Non-Linear Effects of Temperature Circuit

In addition, one feature of the circuit involves the use of trimpots for offset and calibration adjustments. Small 200Ω trimpots provide a means of removing the temperature offsets which occur due to the slight resistance variances between transmission gates and thermistors. Since each of these components are different, slight temperature offsets of one or two degrees could occur if no compensation is provided. Also, larger 20K trimpots can be switched into the circuit in place of the thermistors as shown in Figure 3.15. This provides a means of verifying and calibrating the circuit without the use of thermistors.

3.3 Output

The output section utilizes only one mechanism to output data. This machanism makes use of the display section circuit along with processor software to arrange data into a user comprehensive format.

3.3.1 Display

The display section forms the output portion of the communication link between the user and the machine. All of the apple scab program's accumulated data and predictions are presented to the user via a low power, high contrast, four character alphanumeric LCD and the associated driving circuitry. The display driving circuitry, shown in Figure 3.16 directs the character data transfer, stores the input character data, and outputs the character illuminating signals.

Each execution of the display output routine transfers four 16 segment characters to the display. During this execution, four bits of character selection code route, by way of a 4 to 16 decoders, each ½ character (eight bits) of data to the proper LCD driving chips. This

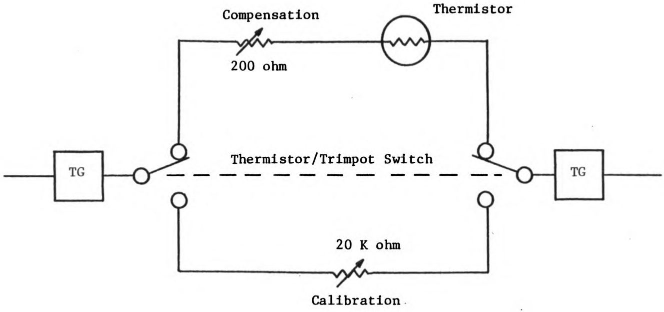


Figure 3.15 -- Detail of Thermistor Circuit

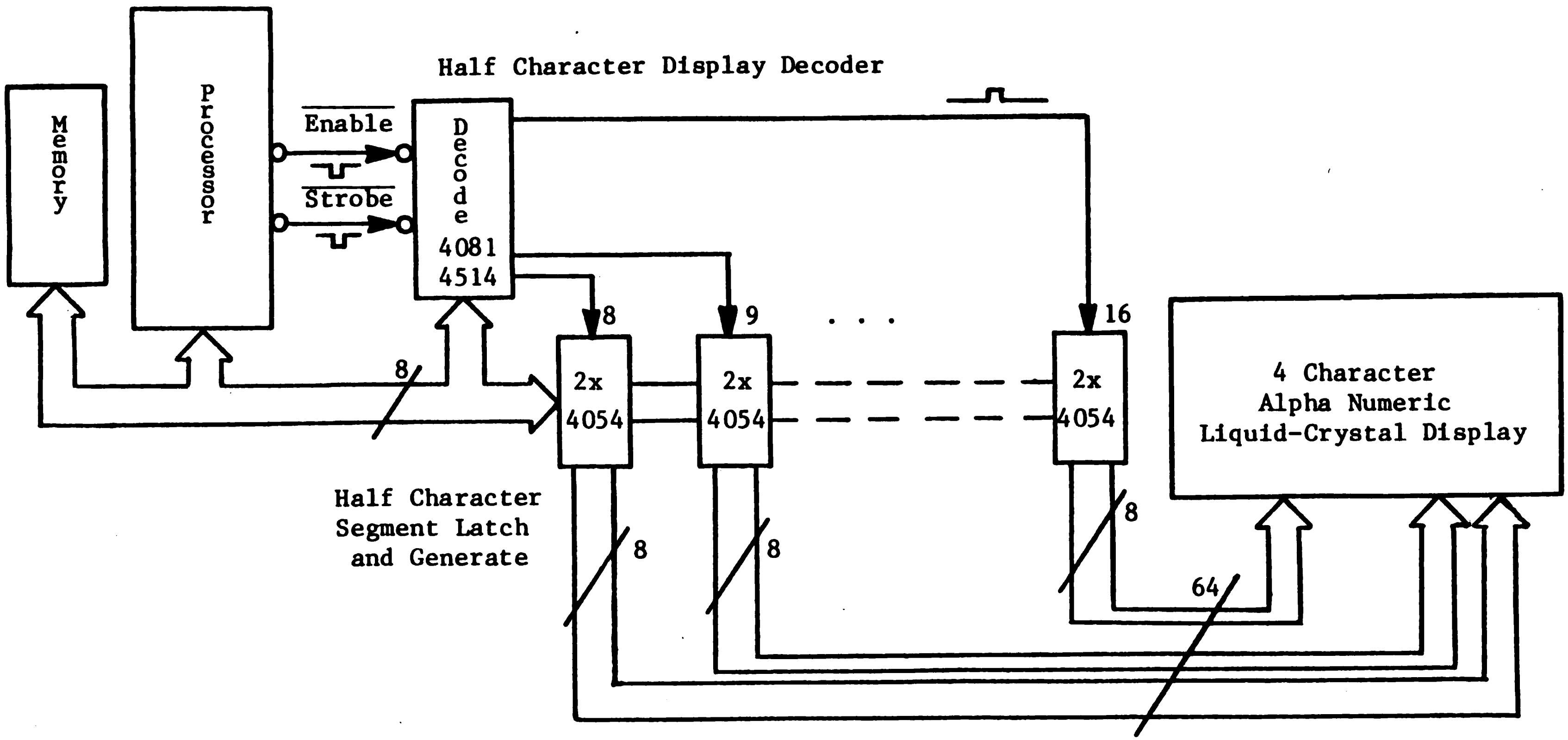


Figure 3.16 -- Display Driving Circuit

character data is then latched into the respective driving chips upon termination of the routing signal. The actual in-phase and out-of-phase square wave LCD segment illumination signals are then produced by the LCD driving chips according to this stored data. The example listed in Table XVIII illustrates the output procedure to output one character.

3.4 Support

The support section provides the ASDP circuit with power, environmental protection, and hardware debug facilities. Three subsections, debug, power and other non-circuit board hardware, (refer to Figure 3.1) provide this support.

3.4.1 Debug

The debug section allows hardware and software to be evaluated through the use of a microterminal and evaluation kit processor board. A complete hardware description of these components can be found in the RCA COSMAC Evaluation Kit Manual.[10]

3.4.2 Power

The power supply circuitry provides continuous regulated power to all of the circuitry, including the debug microterminal and processor board. This performance is made possible by dual low voltage power supplies, two diodes and a regulator. The dual power supplies, a 12 Volt battery at 200 AH and an 8.7 Volt battery at 2.3 AH, in combination with two germanium diodes as diagrammed in Figure 3.17 insure continuous glitch free power. The 12 Volt or main source is responsible for powering the circuit and trickle charging the 8.7 Volt source. During failure, the 8.7 Volt auxillary supply takes over and the user is notified

TABLE XVIII

PROCEDURE TO OUTPUT ONE FULL CHARACTER

1. Output the half character selection number eight to the 4 to 16 decoder, but do not decode it.

2. Output the half character segment data to the LCD driving chips as selected by the decoder (line eight).

3. Repeat Step One except use number nine.

4. Repeat Step Two.

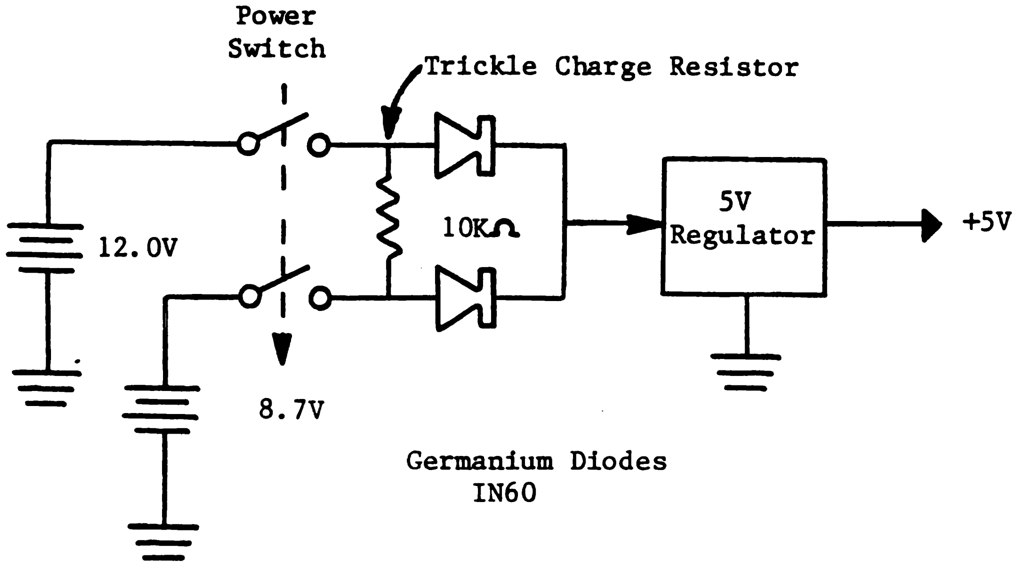


Figure 3.17 -- Power Supply Circuit

by a message-power flag.

In addition, the germanium diodes provide the necessary power supply isolation, yet allow each source to power the circuit with minimal voltage and power loss. The regulator completes the power supply circuitry by regulating the dual source variable voltage inputs to the required flat five Volt circuit supply. Typical power drawn by the entire circuit during normal operating mode is 30 mA at 12 Volts or .36 Watts.

3.4.3 Non-Circuitboard Hardware Components

The non-circuitboard hardware components form the backbone of the predictor machine by supporting and protecting the internal circuitry. The components which provide this support include the enclosure, the wire connectors, and the instrument package support.

The enclosure, a Hoffman Nema type 4 clamp cover box, provides the circuitry with a water and air tight seal. This seal, along with a moisture-absorbing desicant protect the circuitry from condensation and its detrimental effects. The desicant is housed in a holder with humidity indicator which connects through the box to allow visual inspection of the humidity level and desicant condition.

The cover of the enclosure supports the keyboard and LCD. The keyboard fits directly into a rectangular hole cut into the cover and forms a moisture-tight seal with the cover. The LCD is viewed through a plexiglass window in the cover. Both the LCD and keyboard are securely attached to the cover.

The enclosure interior supports various components. Directly in the center of the enclosure is the regulator support hardware. To the left side of the box are the internal batteries which provide the auxillary source of 8.7 Volts. Throughout the interior of the box are the main circuit board supports.

Scotch Flex and Cannon connectors solve the wire connection problem by providing mass termination and a weather-proof seal. The Scotch Flex connectors terminate the 73 LCD and keyboard leads to the circuit board, while the Cannon connectors provide an airtight means of connecting the external sensors and 12 Volt power source with the internal circuitry.

The entire instrument package is supported and protected by a white wooden instrument support. This support, as shown in Figure B.2, shields the instument from sun and rain and houses the temperature sensors.

CHAPTER IV

SOFTWARE ORGANIZATION

The operational character of the ASDP is determined by the apple scab program or software package (illustrated in Figure 4.1). Three functions are encompassed within this software to enable the ASDP hardware to acquire, evaluate, and display data. Three additional functions are also included to facilitate hardware/software initialization, interrupt handling, and hardware/software debug. The hardware/software initialization or start up and wait function, discussed in Section 4.1, occurs first after the ASDP is activated and is responsible for placing the instrument in an operational mode. The occurrence of an interrupt then activates the interrupt service function, Section 4.2, to distinguish which of four interrupt routines must be executed. Two of these four interrupt routines (Clock/Prediction and Rainfall) form the data acquisition function, Section 4.3, of the ASDP and operate to transmit environmental data to the ASDP. The transmitted data is then evaluated through the data evaluation function, Section 4.4. A third interrupt routine (Keyboard) then implements the data display function, Section 4.5, to transfer data and results to the user. Finally, the hardware/software debug function, Section 4.6, incorporates the fourth interrupt routine (Debug) to simplify the task of developing new software and trouble-shooting existing hardware.

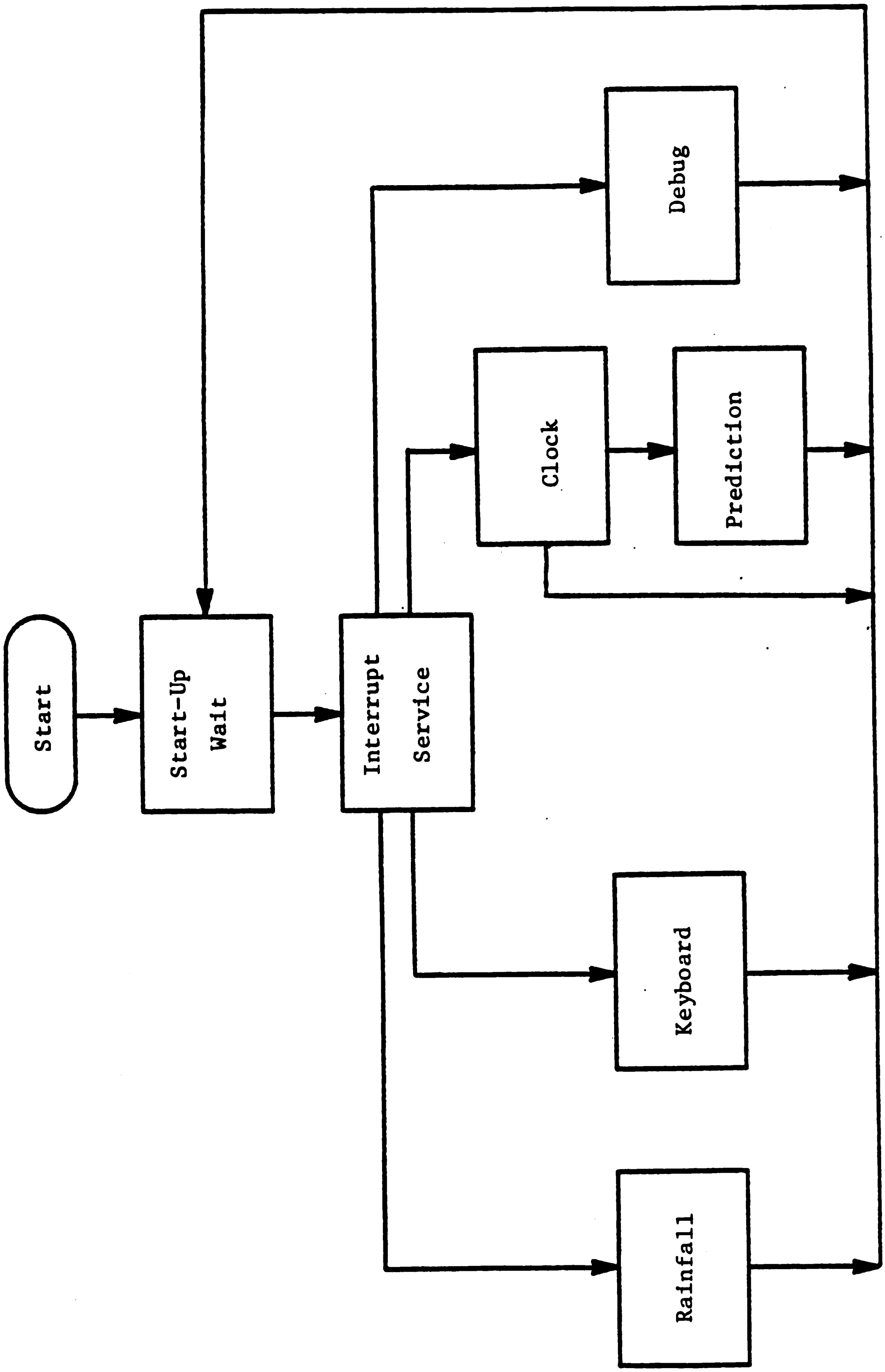


Figure 4.1 -- General Software Organization

4.1 Hardware/Software Initialization

The start-up and wait function performs two operations as suggested by its name. The first of these, start-up, initiates the program after the reset and run switches have been activated properly. This initialization procedure includes setting up the internal microprocessor registers, clearing the prediction storage locations, setting the date, time, dry and wet bulb temperatures, humidity, and leaf wetness (JA01,000,0^{o}F and 0^{o}F, high, and wet), reseting the buffer pointers, and clearing each of the history buffers. Following the start-up operation, or after an interrupt is completed, the wait operation is executed. This operation simply writes a branch-to-itself loop into RAM memory and jumps to it. With this loop in RAM, power consumption is considerably reduced as mentioned earlier because the power consuming ROM are deselected. The microprocessor will then continue to execute this wait branch loop until an interrupt occurs.

4.2 Interrupt Service

The interrupt service function performs all service operations associated with hardware interrupt requests. These operations include saving critical registers upon a stack, directing the hardware to input an interrupt status word, decoding the status word to determine which of the interrupt routines should be activated, outputing a new interrupt mask to disable further interrupts, and directing the microprocessor to one of the interrupt routines (refer to Figure 4.2). Presently, one of four interrupt routines--Rainfall, Keyboard, Clock/Prediction, and Debug--can be activated as determined by the interrupt status word (see Table XIX). The interrupt service function decodes this status

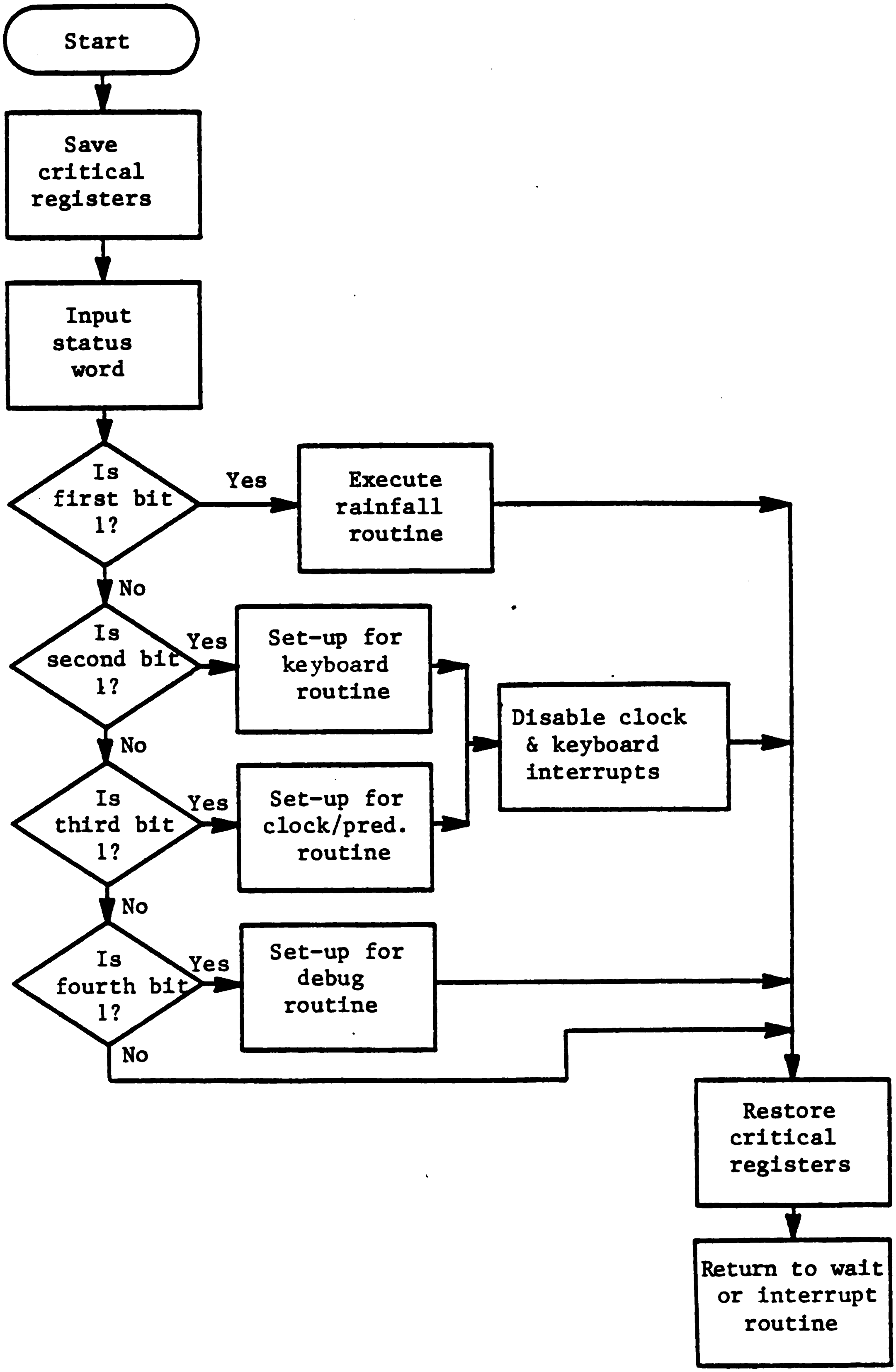


Figure 4.2 -- Interrupt Service Routine

TABLE XIX

INTERRUPT STATUS WORDS

Status Word (msb) (lsb)	Interrupt
XXXX XXX1	Rainfall
XXXX XX10	Keyboard
XXXX X100	Clock/Prediction
XXXX 1000	Debug

(X's represent don't cares)

word by looking for the first bit position which contains a one, reading from right to left. The bit position where the first one is found then specifies which interrupt routine must be executed. An interrupt priority is obviously defined with this to be Rainfall highest, Keyboard second highest, Clock third highest, and Debug lowest.

4.3 Data Acquisition

The data acquisition function receives weather data via the two interrupt routines; Rainfall and Clock/Prediction. The Rainfall routine records rainfall in hundredths of inches by incrementing a counter (microprocessor register) upon the occurrence of each rainfall interrupt (1/100 of an inch of rain). The rainfall data is then stored at ten minute intervals through the Clock/Prediction routine.

The Clock/Prediction routine completes the data acquisition function by recording dry and wet bulb air temperatures, humidity, and leaf wetness. Specifically, the Clock portion of this routine updates the current date and time, and records the current temperature and humidity level. The stepwise operation of this Clock routine,

Figure 4.3, includes reseting the clock interrupt, incrementing the current time, checking for a new day (1440 minutes), checking for a new month (reached last day of month), and checking for a new year (reached December 31). At this point the date and time operation is complete and the temperature calculation is begun. The Clock routine continues by incrementing a temperature timing register (RE.0), determining if this time is odd or even, setting up the temperature circuitry for either wet or dry bulb as specified by odd or even time, triggering the temperature circuitry (number of temperature/DMA cycles are recorded in RO), and summing the temperature/DMA cycles onto a dry or wet bulb storage location. The temperature timing register, RE.0, is now inspected to determine if a ten minute interval is complete. A negative result (less than ten minutes) from this inspection will terminate the Clock routine at this junction, while a positive result will continue the routine toward the computation of average dry and wet bulb temperatures.

The dry and wet bulb temperature computation process involves dividing both the dry and wet bulb total number of temperature/DMA cycles by 20 and subtracting a constant to determine a temperature table location in the range of 0–255. The selected temperature table location then presents the microprocessor with a value that corresponds to twice the actual temperature (refer to Table XX for an example). The microprocessor stores and computes temperatures with this double magnitude value to gain accuracy to within one-half of a degree fahrenheit. On the occurrence of an output operation this double magnitude temperature is divided by two to produce a valid single magnitude temperature for display. The temperatures are software calibrated by

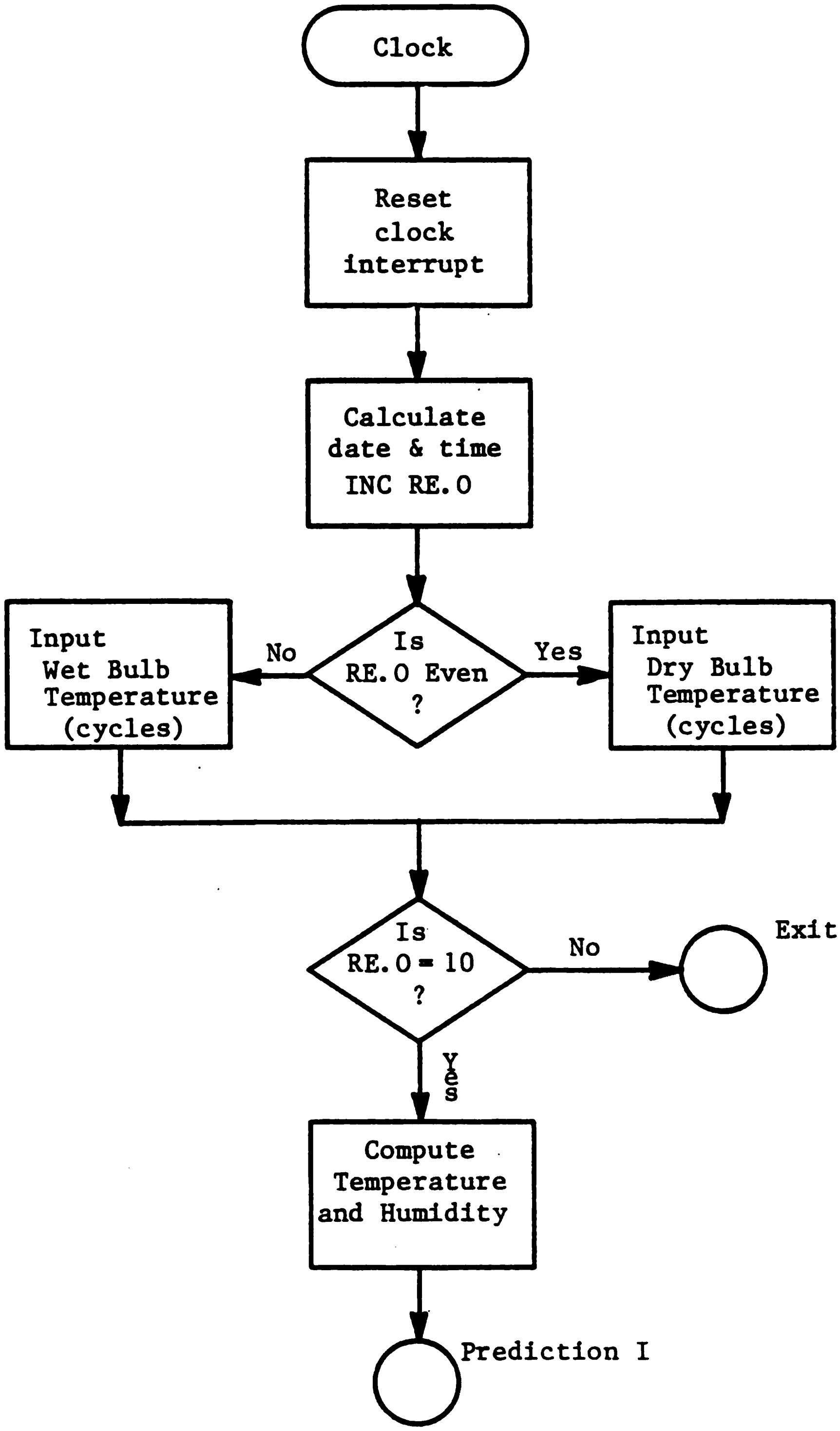


Figure 4.3 -- Clock Routine

TABLE XX

TEMPERATURE CALCULATION PROCESS

Time	Actual Temperature		DMA Cycle Temperature	
	Dry	Wet (F)	Dry	Wet (cycles)
1	70.5		1238	
2		64		1291
3	71		1234	
4		63.5		1295
5	70.5		1238	
6		64		1291
7	72		1226	
8		63		1299
9	71.5		1230	
10		63.5		1295
Totals	355.5	318.0	6166	6471
Averages	71.1	63.6		

Computer Routine (Unit T)

Divide by twenty — Rounding (.5 and above is rounded up)
- Dry 6166/20 = 308.3 — 308
- Wet 6471/20 = 323.6 — 324

Subtract Constant (210)
- Dry 308-210 = 98
- Wet 324-210 = 114

Temperature Table Value (Stored Value)
- Dry 98 yields 142
- Wet 114 yields 127

Display Temperature
- Dry 142/2 = 71°F
- Wet 127/2 = 64°F

tuning the temperature table values and subtraction constant to produce accuracy within $\pm$ one degree F over a temperature range of 0–120°F.

After the temperature calculations, the Clock routine continues by determining if an infection period is in progress and if a thirty minute interval as determined by RE.0 has passed. A positive response to both of these questions places the current dry bulb temperature into location RTEMP for eventual averaging to produce an average air temperature for the entire infection period. On the other hand, a negative response to only the RE.0 interval question places a value of zero into RTEMP. A negative response to the infection in progress question places the current temperature into RTEMP on each occurrence of this sequence (every ten minutes). In summary, this result places the current air temperature into RTEMP every ten minutes if no infection is in progress and every half hour if one is in progress. This procedure allows a temperature to be recorded when an infection period begins, while not allowing excessive temperature data to overflow the average temperature summing location.

The last portion of the Clock routine, finally, calculates the humidity level by determining if the current dry and wet bulb temperatures differ by less than three degree fahrenheit. A high humidity (greater than or equal to 90%) result is recorded if the temperatures differ by less than three degrees F, while a low humidity (less than 90%) result is recorded if the temperatures differ by three degrees or more. These two humidity levels are recorded as three for high and one for low to simplify the display routine.

The Prediction portion of the Clock/Prediction routine completes the data acquisition function by determining if the leaves are wet.

This simply involves testing a flag and storing the result as four for wet and five for dry, again for ease of display.

4.4 Data Evaluation

The data evaluation function processes the input weather in accordance with Jones Alogrithm (Section 2.3) to predict the likelihood of apple scab formation. The computed results along with the input weather conditions are then stored for later display to the user. The Prediction portion of the Clock/Prediction routine totally encompasses and partitions the data evaluation function into two routines; Prediction I, determination of an infection period, and Prediction II, determination of a scab severity level.

Prediction I determines the occurrence, length, and termination of an infection period by selectively controlling the amount of time which contributes toward it. This selective process is accomplished by utilizing the past and present weather data to choose one of ten current infection states or paths of operation (see Table XXI). This operational path then dictates the operations to be performed to update the infection (see Table XXII). As an example: If it suddenly started to rain after two days of dry low humidity weather, Prediction I would select State A (new infection period) as the path of operation. The operations then performed as directed by this path include: advance short buffer (6) one unit (16 data locations); clear the old data in buffer 6; store the current date and time into buffer 6; increment the total amount of time and the wet leaf time for this infection to ten minutes; store the wet leaf time and the rainfall amount into buffer 6; and branch to Prediction II routine. The entire layout

TABLE XXI

CURRENT INFECTION STATES

State	Conditions
A	Wet Leaves, First Occurence (New Infection Period)
B	Wet Leaves, Following Wet Leaves, Scab=None
C	Wet Leaves, Following Wet Leaves, Scab≠None (Low,Mod,High)
D	Wet Leaves, Following Dry Leaves, Scab=None
E	Wet Leaves, Following Dry Leaves, Scab≠None (Low,Mod,High)
F	Dry Leaves, No Infection in Progress
G	Dry Leaves, High Humidity, Scab=None
H	Dry Leaves, High Humidity, Scab≠None (Low,Mod,High)
I	Dry Leaves, Low Humidity, Scab=None
J	Dry Leaves, Low Humidity, Scab≠None (Low,Mod,High)

of Prediction I software is illustrated in Figure 4.4. As shown in the figure all Prediction I sequences terminate on one of two paths, exit or Prediction II. Each exit path terminates the routine while each Prediction II path directs the microprocessor to the Prediction II routine.

The Prediction II routine determines the scab severity level from an infection period (developed by Prediction I) and an average air temperature for the infection period. The stepwise procedure of the Prediction II routine (Figure 4.5) includes, first determining if the scab severity should be calculated. This involves testing to see if the total period of time in which this infection period is occurred greater than or equal to five hours. A negative test result causes the routine

TABLE XXII

OPERATIONS TO UPDATE INFECTION PERIOD

State	Operations
A	Store current date and time into buffer 6 Increment Total and Wet Leaf Store Wet Leaf and Rain in buffer 6 Branch Prediction II
B	Increment Total and Wet Leaf Store Wet Leaf and Rain in buffer 6 Branch Prediction II
C	Increment Total and Wet Leaf Branch Prediction II
D	Increment Total and Wet Leaf Add Wait time onto Total, zero Wait Store Wet Leaf, Dry Leaf, High Relative Humidity, and Rain into buffer 6 Branch Prediction II
E	Increment Total and Wet Leaf Add Wait time onto Total, zero Wait Branch Prediction II
F	Branch Exit
G	Increment Total and High Relative Humidity Add Wait time onto Total, zero Wait Store Dry Leaf, High Relative Humidity, and Rain into buffer 6 Branch Prediction II
H	Increment Total and High Relative Humidity Branch Exit
I,J	Increment Wait and Dry Leaf Check Dry Leaf to establish if termination requirements are met (8 hours None, 6 hours Low, 4 hours Moderate, 4 hours High) No, Branch Exit Yes, Reset Prediction Loc. Branch Exit

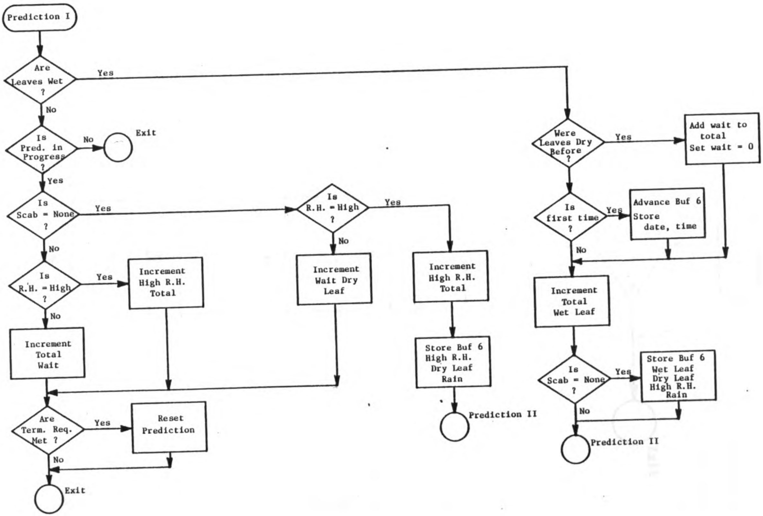
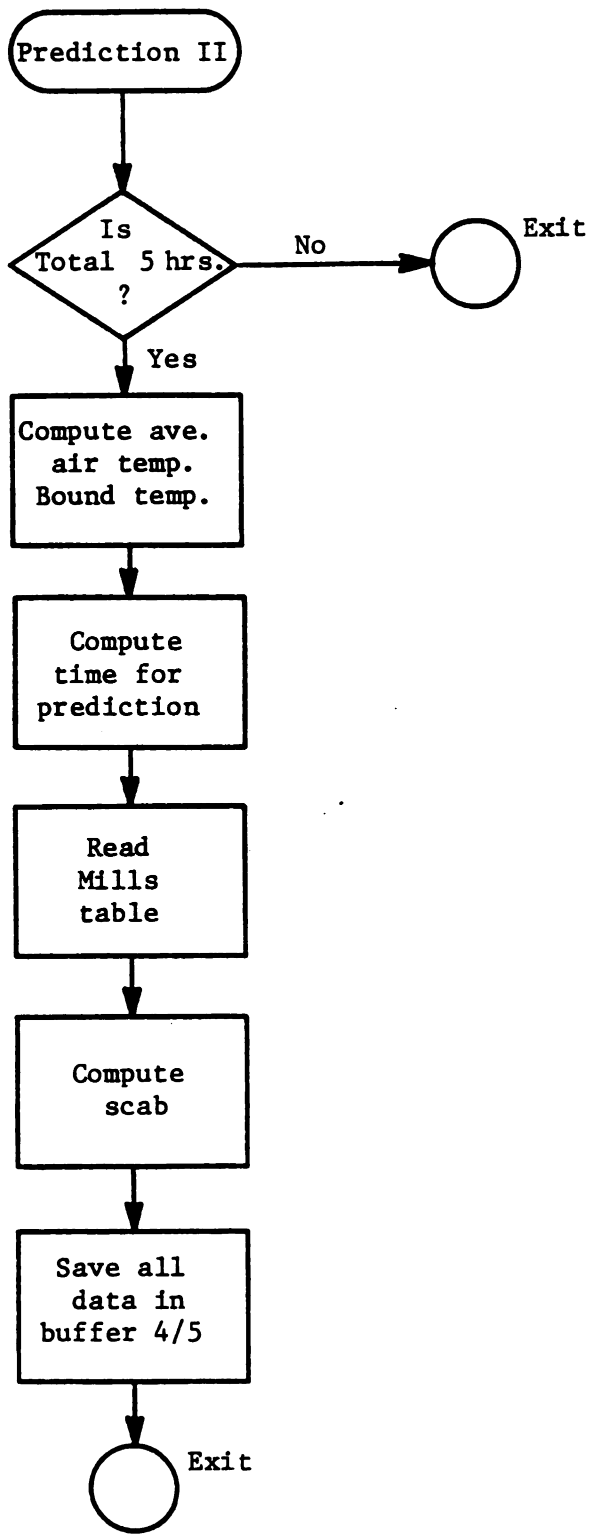


Figure 4.4 -- Prediction I Routine



Rigure 4.5 -- Prediction II Routine

to terminate, while a positive result leads to a second test in which the number of times through this routine is questioned. A result to this second test of zero times then directs the set up of buffer 4/5 and the subsequent storing of infection date and time. With the result of the above test being greater than zero or after buffer 4/5 set up, the Prediction II routine continues by calculating an average temperature for the infection period. This temperature is achieved by directly averaging the stored sum of half hour temperature data which had been collected throughout the total time period in which an infection period is compiled. The temperature is then conditioned to insure that it does not exceed the bounds of the scab table (33-78^{o} F).

The infection period (total time leaves are wet, plus all times leaves are dry with high humidity and none scab severity level, plus all dry leaf periods of time enclosed by either wet leaf or high humidity and dry leaf occurrences with none scab severity level--total wet leaf time, plus dry leaf and high humidity times stored in buffer 6) is then converted into half hour units and compared with Modified Mills Table values for the calculated average temperature to yield a scab severity level (see example Table XXIII). This new scab severity level is then compared with the previous level to determine which is highest. The highest level is finally stored as the predicted level of scab severity. The closing sequences of the prediction routine then store calculated temperature and scab severity level along with most of the weather data into the long buffer (4/5) for later display to the user.

4.5 Data Display

The data display function transfers the input weather data and

TABLE XXIII

PREDICTION II EXAMPLE

Condition		Duration
Wet Leaf		8.00 (hours)
Dry Leaf (buffer 6)		2.00
High Relative Humidity (buffer 6)		4.00
	Total	14.00

Infection Period = 28 half hours
Average Temperature 68°F

Modified Mills Table (excerpt)

Temp	Low	Moderate	High
68 F	18	24	36 half hours

Computed Scab Severity = Moderate

evaluated results to the user via the Keyboard interrupt routine. This operation is completed by three divisions of this routine: control; display activity; and display sequence. The Keyboard control division is responsible for interpreting what information the user desires by the sequence of keys he depresses on the keyboard. Upon depressing the set up key (#) control division understands that the next numerical key (0-9) is the command which the user desires to execute. Each command contains a sequence of display activities of which only one is executed upon each depression of the continue (*) key. The sequence of display activities for each command are stored in the display sequence section and function as a look up table as to what must happen next. The true elements of the data display function are then the display activities.

Each of the nineteen display activities performs a specialized operation which results in displayed information of some sort. A partial discription of each of these activities or display routines is listed in Table XXIV.

TABLE XXIV

KEYBOARD DISPLAY ACTIVITIES

Routine	Description
1	Displays Words (four alphanumeric characters)
2	Displays Rainfall in hundredths of inches
3	Displays Times in hours and minutes
4	Displays Dates in month and date format
5	Displays Temperatures in degrees fahrenheit
6	Displays Scab Levels, Humidity Levels, Leaf Wet Levels
7	Displays XXXX and sets machine to receive password
8	Checks the passwords
9	Resets the prediction
10	Checks input dates and times
11	Sets up each buffer 4/5 history unit (16 locations) for display
12	Sets up each buffer 6 history unit (16 locations) for display
14	Displays bbb + (b = blank)
15	Calculates Time until infection
16	Set up routine for Eradicant Display
17	Calculates time to apply Eradicants
18	Set up for number of sprays
19	Set up routine for rainfall density
20	Sets up each Rainfall density history for display

Other features of the Keyboard routine include the ability for it to accept numbers, when they are required by the command being executed, and the ability to recognize illegal key sequences. A partial software flowchart is listed in Figure 4.6.

4.6 Hardware/Software Debug

The major operations of the debug interrupt routine include memory and internal register inspection and modification, program resumption with start point selection, and hardware execution. These operations comprise the debug function, the last function of the ASDP. Figure 4.7 illustrates the software structure which implements these routines. Four command keys--$P, ↔, CA, INC--on the microterminal direct the entire operation of this routine. The $P command directs the microprocessor to return to the main apple scab program via an interrupt return. Program execution resumes with all registers restored at the point where it ended or at a new location specified by the user. The ↔ command switches the location of the decimal points from the right four digits to the left four digits of the microterminal and visa versa. Their position then informs the Debug program that the decimal portion of the display is not to be modified by incoming digit data. The CA command steps to and enters the next one of five possible modes (Display/Modify Memory, Display/Modify Registers, Display/Modify DF-XP-D Output to 4514, and Output to 4508). Upon entering a selected mode a pointing register (R8), which points to where the mode data is stored, is initialized. This register is then utilized in various ways to fill an 8-byte display character storage (8 bytes of RAM). With this last command the CA command is completed and the operation of the microprocessor is returned to the display multiplexing and keyboard polling routine. A numerical keyboard input (0-F) then modifies the pointing register

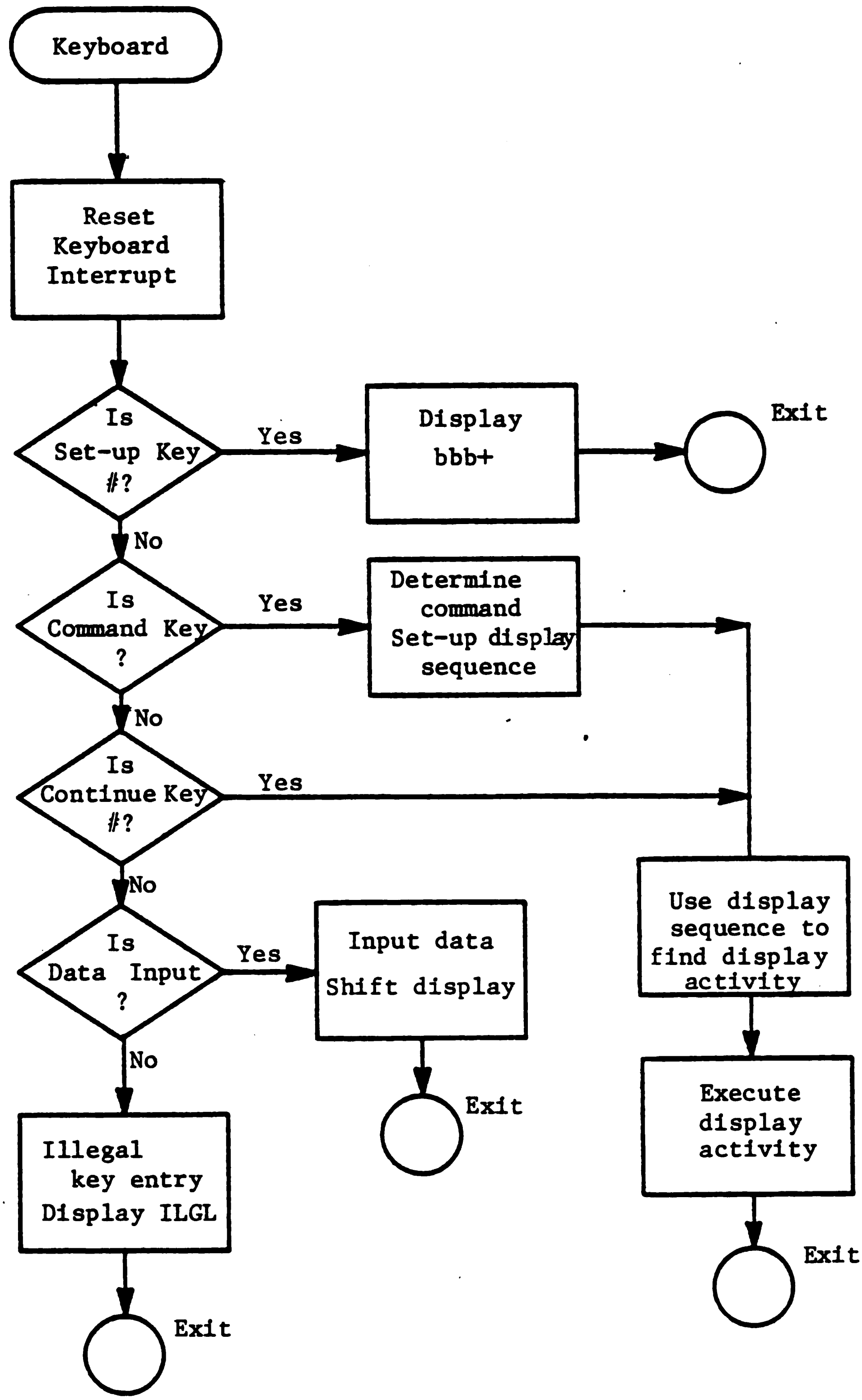


Figure 4.6 — Keyboard Routine

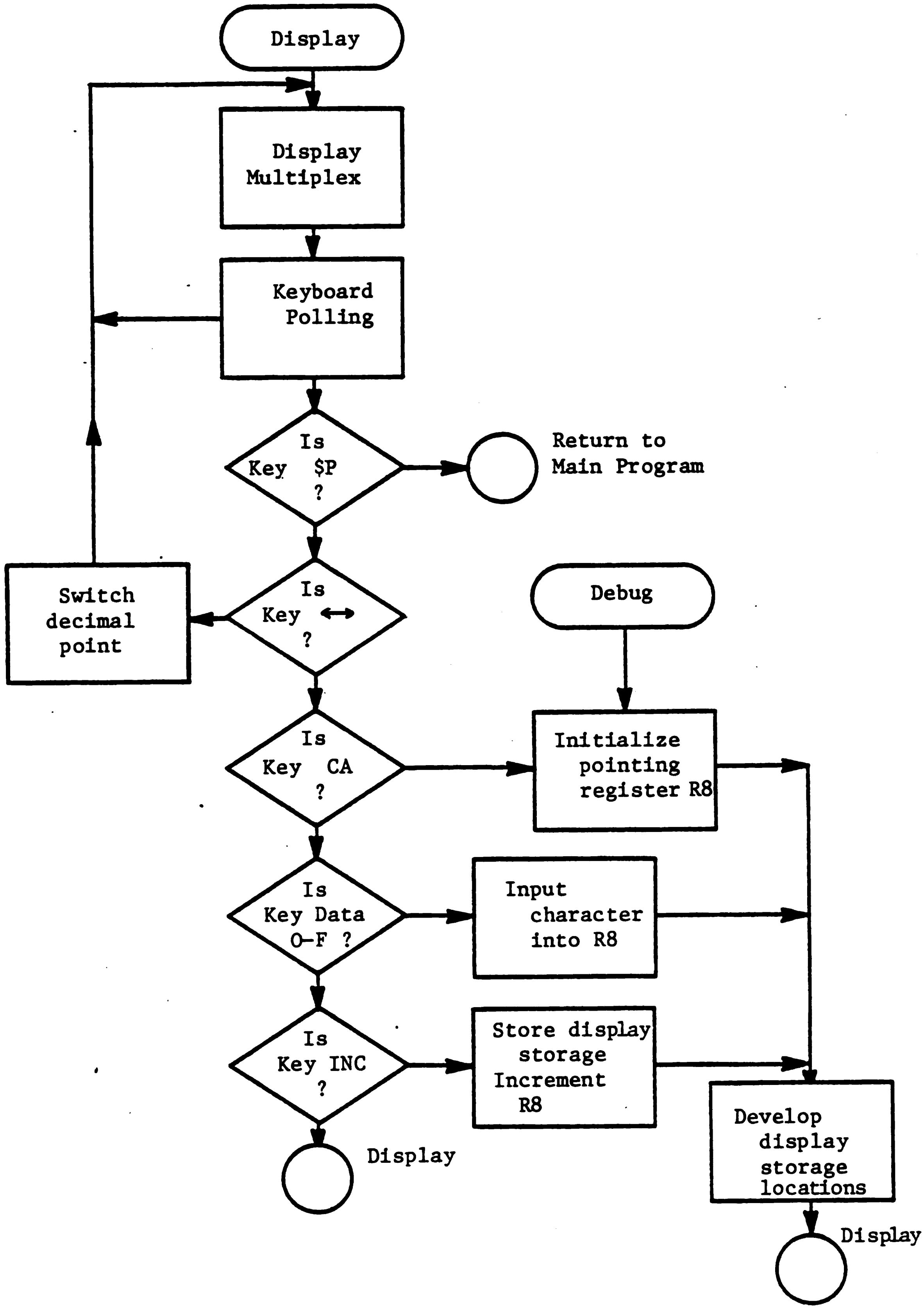


Figure 4.7 -- Debug Routine

(decimal to left) or the display character storage locations (decimal to right) dependent upon which type of memory is protected. Finally, the INC command either decodes the 8 bit display memory, stores it via the pointing register, and increments the pointing register, or activates the hardware as specified by the display memory.

CHAPTER V

DEVELOPMENT AND EVALUATION

This chapter presents an outline of the procedures used to develop and evaluate the ASDP. And results of this evaluation are summarized. Each of the sections of this chapter focus on one of the following phases of development; study (Section 5.1), design (Section 5.2), construction (Section 5.3), software development (Section 5.4), and field evaluation (Section 5.5). These phases have played a major role in the development of the ASDP and represent a substantial portion of the overall development time.

5.1 Study

The study phase focused on developing an understanding of apple scab and learning the RCA microprocessor. This required approximately two man-weeks (40 hour week) devoted to the study of apple scab for the purpose of determining how to record the necessary environmental data and implement the scab routine. Specifically, this portion of the study dealt with defining the sensors needed, the states of the scab routine, the user interface functions, and the physical considerations of the instrument for the environment in which it must operate. Similarly, approximately three man-weeks were devoted to the tasks of studying the general architecture, simple programs, and hardware configurations for the microprocessor. Through this study many of the software and hardware problems associated with interfacing the microprocessor with any instrument were examined. A few of the software problems included interrupt handling, subroutine branching, and mathematical routine implementation (multiplication, BCD conversion, and addition). Similarly, hardware configurations focused on the problems associated with interrupt debounce, memory addressing, and input/output control. Upon the completion of this study phase the microprocessor had been thoroughly studied and the design phase begun.

5.2 Design

The predictor design phase contrary to the study phase dealt specifically with designing the hardware architecture of the ASDP. Preliminary work included both studying a progenitor and consultation with the designer, Sig Lillevik, to determine the features to be incorporated into the ASDP.[11] Other preliminary work involved deciding hardware versus software tradeoffs. A few of the specific functions involved with this included interrupt debounce and decoding, temperature measurement, and leaf wetness measurement. The results of this preliminary work (1 man-week) were then incorporated into the ASDP hardware design. The total design process (including preliminary work) required two iterations and lasted 1 man-month before a third and final hardware design was settled upon. Parts were then ordered and construction begun.

5.3 Construction

The construction phase lasted approximately two man-months. During this time the entire unit was constructed and tested. Construction was segmented into four tasks. These included circuit board, display and keyboard interface, enclosure support, and sensor construction. Testing was performed at the conclusion of each of these tasks to verify their operation. Some of this testing was augmented by the use of short microprocessor test programs. These test programs were significant in that they placed the machine in an operational mode and provided actual data as to the performance of circuit sections.

5.4 Software Development

The software development phase was the final phase of construction of the ASDP. Four tasks segmented this development and simplified the overall process. These were flowchart development, flowchart to code translation, code assembly and simulation, and code execution upon the ASDP. Of these tasks, code assembly and simulation required the most time, one man-month. On the other hand, it yielded the most information as to if a specific routine was functionally correct. This code assembly and simulation task was complete by way of the MSU time-sharing and the RCA COSMAC cross assembler and simulator. Code execution upon the ASDP hardware then acted as the final test for proper software operation. After two man-months of software development, the ASDP program was completed and the unit was readied for field evaluation.

5.5 Field Evaluation

The field evaluation began on August 25, 1978 when the two ASDP units (unit T and unit K) were installed in the MSU research apple orchard. During the following months at least one of the two units was in the field until November 16, 1978. Over this time span, the ASDP's operation was nearly perfect. Three small problems occurred during the evaluation period. The first occurred on September 25 when unit K scrambled the stored data when the temperature dropped below 44°F. As temperature dropped, two of the RAM socket pins became disconnected from the RAM, causing data to be misinterpreted. Re-soldering this intermittent connection solved this problem. A second problem occurred on October 5 when unit K's wet bulb temperature rose three degrees above its dry bulb temperature. The problem was diagnosed to be caused by the thermistor element. Water had penetrated the thermistor protection and corroded the element in such a way so as to raise the temperature by three degrees. This effect was compensated by the addition of resistance into the circuit by way of a 500 ohm trimpot in place of the 200 ohm compensation trimpot (refer to Figure 3.15). Another problem involving both units was due to slight errors in the software. This frequently occurred after new routines or programs were implemented. Although most corrections were reasonably simple software modification was still a time-consuming process.

Despite minor problems, the ASDP functioned exceptionally as exemplified in the following comparisons that compare recorded meteorological data with the equivalent ASDP data. The categories in which these comparisons were made include temperature (dry and wet bulb), leaf wetness, rainfall, and scab infection period. Dry and wet bulb temperature measurements were usually within 1 F of those recorded by a strip chart recorder and sling pychrometer (see Table XXV). Atypical data occurred on cool bright sunny days where the sunshine heated the temperature sensors enclosure. This result is denoted by the words "bright sun" in the table. A modification in the sensor enclosure is in the design phase to reduce this sunshine heating effect.

TABLE XXV

TEMPERATURE MEASUREMENTS

		Instrument		Alternate Method	
Date	Time	Dry	Wet	Dry	Wet
OC12	1942	54 F		54 F	
OC16	1731	49	43 F	48	42 F
OC17	2136	40		40	
OC18	1736	54		53	
OC19	1917	50		50	
OC20	1644*	70	59	68	56*
OC22	1059	67	60	66	59
OC23	1640	48	41	47	42
OC24	1650	42		41	
OC25	1326	49		49	
OC29	1540*	61	52	53	45*

*Bright Sun

Alternate Method
Dry Temperatures from Strip Chart Recorder
Wet Temperatures from Sling Pychrometer

TABLE XXVI

LEAF WETNESS MEASUREMENTS

		Hours of Wet Leaves	
Date	Time	Instrument	deWit Leaf Wetness Recorder
SP15	2304	11.66 hours	11.00 hours
SP17	454	6.66 hours	6.00 hours
SP18	2404	11.50 hours	12.00 hours
OC22	2344	10.50 hours	10.30 hours
OC25	1314	21.50 hours	21.00 hours
NV01	943	1.00 hours	1.00 hours
NV02	633	2.66 hours	3.00 hours

The ASDP leaf wetness sensor typically recorded leaf wetness periods within 40 minutes of those recorded by the DeWit leaf wetness recorder. This 40 minute accuracy is considered quite satisfactory for a leaf wetness sensor. A sample of these recordings and their comparisons are listed in Table XXVI. On the other hand, the ASDP leaf wetness sensor did not record many heavy dew wetting periods. This was unsatisfactory and was subsequently corrected by reducing the wetness threshold of the leaf wetness circuit.

The rainfall totals recorded by the ASDP were fairly accurate as listed in Table XXVII. Note however, that the ASDP records were occasionally lower than those recorded by the alternate raingage method. This is believed to be due to heavy dew collection in the raingage before a rainfall. Currently no method except emptying the raingage before a rainfall has been developed to remove this effect.

TABLE XXVII

RAINFALL RECORDS

Date and Time of Measurement		Rainfall in Hundreths of Inches	
Date	Time	Instrument	Alternate Method
SP27	1536	16	18
SP28	1044	17	19
SP30	1851	6	7
OC03	1741	25	26
OC04	1818	4	4
OC05	1522	14	14
OC16	1731	79	81
OC23	1644	32	34
OC25	2035	20	21
OC26	1326	15	15

The scab total infection period computed by the ASDP is accurate when compared with the length of total infection periods calculated by hand. Table XXVIII which compares both strip chart data with the ASDP data illustrates this point. These infection periods, determined by the scab routine, then lead to the prediction of a scab severity level of fungus growth. At this time there is no data to establish how well the routine performs. Earlier studies with the progenitor do indicate that a similar routine works remarkably well. This fact coupled with the sensor accuracy should lead to precise apple scab severity level predictions.

TABLE XXVIII

TOTAL INFECTION PERIODS

First Period

Strip Chart Recorded Weather Data

Infection period began SP21 at 4:30 (24 hour clock)
Wet leaves from 4:30 to 14:00 (9 hours, 30 minutes)
Dry leaves from 14:00 to 21:00 (7 hours)
Infection period ended

Temperatures	4:00	6:00	8:00	10:00	12:00	14:00	
	70	70	62	56	55	58	(°F)

Temperature average = 61.8°F

Instrument Interpretation

Infection period began SP21 at 4:33
Wet leaves for 9 hours 20 minutes
Infection period ended
Temperature average = 62°F

Second Period

Strip Chart Recorded Weather Data

Infection period began SP20 at 12:40
Wet leaves from 13:00 to 16:00 (3 hours)
Dry leaves from 16:00 to 19:00 (3 hours)
High Humidity from 19:00 to 10:00, NV01 (15 hours)
Dry leaves from 10:00 to 20:00 (10 hours)
Infection period ended

Temperatures	13:00	15:00	17:00	19:00	21:00	23:00	
	62	60	64	62	56	53	(°F)

Temperature average = 58.4°F

Instrument Interpretation

Infection period began SP21 at 12:53
Wet leaves for 3 hours, 40 minutes
Dry leaves for 3 hours, 10 minutes
High Humidity from 6 hours, 20 minutes
Infection period ended
Temperature average = 59°F

CHAPTER VI

CONCLUSION

The purpose of this research effort was to develop, design, construct, and evaluate a microprocessor-based instrument to predict apple scab producing fungal growth. This instrument was to be mounted in a representative orchard location to monitor the weather for the purpose of determining the likelihood of scab infection outbreak. The representative location was to provide the natural meteorological conditions for accurate scab level predictions. Similarly, this location was not necessarily going to provide the indoor conveniences of regulated power and a stable environment. In addition, the unit was to be designed as a prototype of a marketable instrument. This required that ease of set-up, usage, maintenance, repair, and reconfiguration for upgrade or implementation of new functions be considered.

A scab severity prediction and data display routine were the two main software algorithms that had to be developed. The prediction routine was to compile an infection period based on current and past weather states. The resultant infection period was then to be applied to a scab severity table to predict a new level of scab infection. The data display routine was then required to display this contrived level of severity to the user upon proper activation of an instrument front panel mounted keyboard and liquid crystal display. Other information that the user was to receive from this display keyboard interface included current date and time, current weather conditions, current apple scab predictions, hours remaining to apply any one of five different fungicides to avoid crop damage, and a history of the parameters associated with the last sixteen infection periods. Finally, two identical instruments were to be built and evaluated in a research apple orchard for the entire growing season.

Each of these objectives have been met and completed. Currently (April 1979), one unit is located at Cornell University Experimental Station in Geneva, New York, while the other is located in the Botany research apple orchard here at MSU.[12] Performance of both of the units has been outstanding with only minor difficulties as described in Chapter V. Figure B.1 illustrates the field mounting arrangement.

Cost for each unit for parts averages around $970. This can be divided into the various component costs listed in Table XXIX. Further modifications could be implemented on the current design to reduce the unit cost and improve overall performance. Specifically, the chip count can be reduced by incorporating larger (state of the art) memory chips and by employing alternate methods to implement interrupt handling and liquid crystal display driving. This reduction should decrease the cost and construction time of the unit. Additional savings could also be brought about through the use of cheaper wire-wrap sockets and plastic enclosed integrated circuits. These changes should not degrade the instrument operation. The overall performance of the instrument could be improved through the development of a new leaf wetness sensor which would simulate the wetting properties of the leaf more accurately. Finally, the calibration and performance of the temperature measurement operation could also be improved through the use of a temperature stable

TABLE XXIX

COMPONENT COSTS

Microprocessor Memory 1800 Chips	$186.18
All other IC's	114.06
LCD display	32.00
Keyboard	24.00
Augat sockets	68.20
Flat cable connectors	37.25
Crystals	14.65
Five volt regulator	13.50
Precision capacitor	5.00
Round cable connectors	57.32
Switches	14.84
Batteries	72.80
Enclosure	21.94
Flat cable	7.12
Round cable	6.80
Fenwal thermistors	25.72
Capicitors, diodes, resistors	5.86
	$707.24
Tipping bucket rain gage	200.00
Leaf wetness sensor	5.00
	$912.24
Miscellaneous hardware	50.00
TOTAL INSTRUMENT COST (excluding labor)	$972.24

circuit. In general, research with this instrument will continue in an attempt to further optimize hardware and software. Simultaneously, new instruments will be developed to control other orchard pests.

FOOTNOTES

FOOTNOTES

1. Jones in reference 1, page 2, lists the days of incubation vs. temperature.

2. The Modified Mills Table, developed by Dr. Alan L. Jones (Department of Botany and Plant Pathology, MSU), is an extension of the Mills Table. Reference 1, page 2 and reference 2.

3. Dr. Alan L. Jones (Department of Botany and Plant Pathology, MSU) developed the prediction algorithm. Refer to Appendix A.1.

4. The RCA Evaluation Kit (CDP18S020), referred to as a processor board, is essentially a miniature computer minus a means of input. By coupling the kit with a teletypewriter, microterminal (CDP18S021), or other type of input/output device the computer system is complete. Refer to reference 3 and 4 for complete descriptions.

5. Refer to reference 5.

6. Support chips refer to the CDP 1800 series chips. Refer to reference 6.

7. The cross-assembler simulator uses a host timesharing computer system to assemble RCA source code and then simulate microprocessor operation using the assembled object code. Refer to reference 7.

8. Refer to references 3 and 4.

9. Refer to reference 8.

10. Refer to references 3 and 4.

11. Sig Lillevik, a professor at Utah State University, was a doctoral student at Michigan State University at this time. Refer to reference 9.

12. Both units are under the care of the Botany Research teams of the respective universities.
    Dr. Alan Jones -- Michigan State University
    Dr. Robert Seem -- Cornell University

APPENDIX A

APPLE SCAB PROGRAM TWO

A.1 -- Apple Scab Routine Two (DNPDR2)

Statements of Routine

a) Apple scab severity is divided into four levels: none, low, moderate, and high.

b) Routine distinguishes between three meteorological conditions or states.

I	Wet Leaves and High or Low Humidity
II	Dry Leaves and High Humidity
III	Dry Leaves and Low Humidity

c) Infection period refers to the time which is used to compute scab severity. Total period refers to the total time in which an infection period occurs.

d) An infection period begins upon the occurrence of State I or wet leaves.

e) The infection period until "low" level severity consists of the total duration of time the weather is in States I, II, III.

Note: State III is summed onto the infection period only if it is preceeded and followed by States I or II and if the time in State III does not violate the termination requirement.
(Example: Any dry period will be added onto the infection period if it is less than 8 hours and the scab severity is "none".)

f) After reaching "low" level scab severity, the infection period can only be extended by the occurrence of State I or wet leaves.

g) The infection period is terminated upon the passing of X continuous hours of State III. X is determined by current scab severity level of the machine.

Scab Severity	X
None	8
Low	6
Moderate	4
High	4

h) State II will not terminate or extend the infection period once the machine reaches "low" infection.

i) Average temperature is computed throughout the total period regardless of the meteorological state or scab severity levels.

j) Average temperature and infection period duration are then applied to a table to determine scab severity.

> Note: Once a higher scab severity level is reached, the severity cannot be decreased to a lower level.

k) Scab severity status is examined each 10 minutes.

A.2 -- Apple Scab Routine Two

Keyboard Sequence

Function	Button	Display
Calender	1	Current Date Current Time
Scab	2	Scab Severity Level Time until Infection
Sensors	3	Air Temperature Wet-Bulb Temperature Humidity 90% Leaf Wetness Rain in 1/100 of Inches
History Down	4	Date of Infection
History Up	5	Time of Infection Average Temperature over Entire Period Scab Severity Level Total Period Time Wet Leaf Time Dry Leaf Time High Relative Humidity Time Rainfall in 1/100 of Inches
History until Low Level	6	Date of Infection Time of Infection Average Temperature until Low Level Partial Wet Leaf Time Partial Dry Leaf Time Partial High Relative Humidity Time Partial Rainfall in 1/100 of Inches
Eradicants (Back-action sprays) ERD1	7	Date of Infection Period Start Time of Infection Period Start 1-5 Sprays followed by the time remaining for the eradicants to be applied.
ERD2		Date of Infection Period Reaching Low Time of Infection Period Reaching Low 1-5 Sprays followed by the time remaining for the eradicants to be applied.
Prediction Stop	8	PW XXXX=1234 (XXXX must be reached before the password number can be entered) PRED STOP=STOP (curtails current infection period)
Rain	9	Date of 3/100 inches of rain or more occurring in 10 minute interval. Time of 3/100 inches of rain or more occurring in 10 minute interval.

Apple Scab Routine Two (Continued)

Function	Button	Display
Set Calender	0	PW XXXX=5678 DATE JA01=0101 TIME 1200=1200

APPENDIX B

INSTRUMENT PICTURES

Figure B.1 -- Apple Scab Disease Predictor in Field

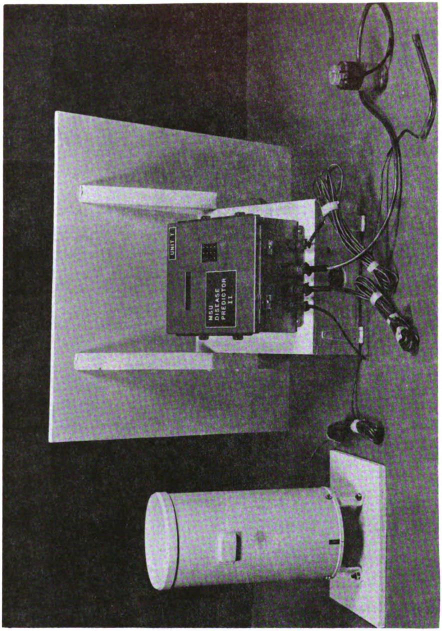


Figure B.2 -- Entire ASDP Package Including Sensors

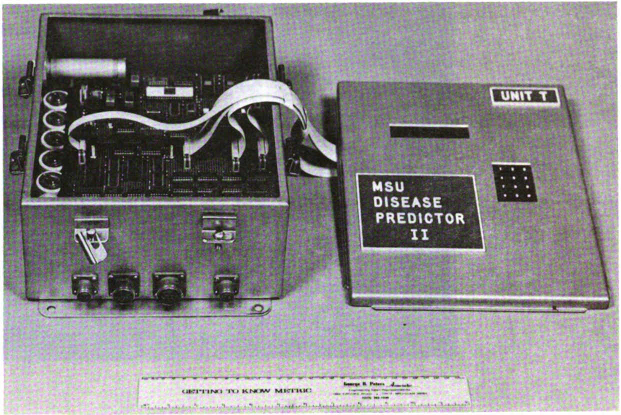


Figure B.3 -- ASDP with Cover Open

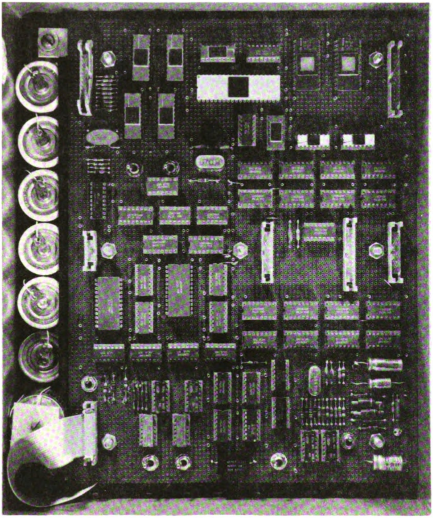


Figure B.4 -- Circuitboard of ASDP

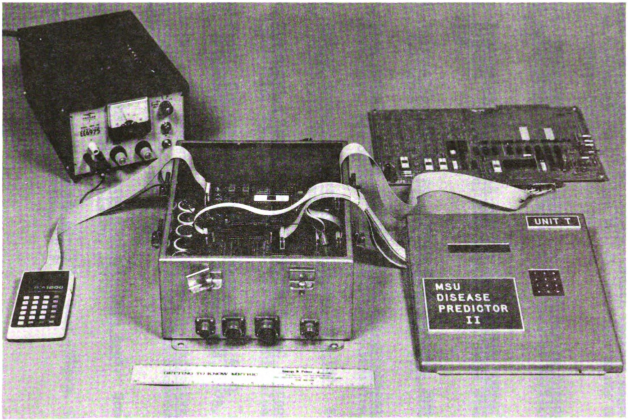


Figure B.5 -- ASDP in Debug Arrangement

REFERENCES

1. Alan L. Jones, Diseases of Tree Fruits. North Central Regional Agricultural Extension Publication No. 45, Michigan State University, East Lansing, October 1976, pp. 1-4.

2. W. D. Mills, Efficient Use of Sulfer Dusts and Sprays During Rain to Control Apple Scab. Cornell University Extension Bulletin, 630, 1944, pp. 1-4.

3. Anonymous, Evaluation Kit Manual for the RCA CDP1802 COSMAC Microprocessor. MPM-203. Somerville, New Jersey: RCA Solid State Division, 1976.

4. Anonymous, Instruction Manual for COSMAC Microterminal. MPM-212. Somerville, New Jersey: RCA Solid State Division, 1977.

5. Anonymous, User Manual for the CDP1802 COSMAC Microprocessor. MPM-201. Somerville, New Jersey: RCA Solid State Division, 1976.

6. Anonymous, COSMAC Microprocessor Product Guide. MPG-180. Somerville, New Jersey: RCA Solid State Division, 1977.

7. Anonymous, Timesharing Manual for RCA CDP1802 COSMAC Microprocessor. MPM-202. Somerville, New Jersey: RCA Solid State Division, 1976.

8. Anonymous, Manual for Tipping Bucket Raingage Model 6010. West Sacramento, California: Weathertronics Incorporated, 1978.

9. S. L. Lillevik, P. D. Fisher, and A. L. Jones, A Predictive Field Instrument for Agricultural Production. Proc. IEEE Microcomputer '77 Conference, pp. 137-142, April 1977.

10. Anonymous, RCA CMOS/MOS Integrated Circuits Data Book. Somerville, New Jersey: RCA Solid State Division, 1977.

11. Anonymous, National CMOS Data Book 62-1375. Santa Clara, California: National Semiconductor Corporation, 1978.