

A11102 461236

NATL INST OF STANDARDS & TECH R.I.C.



A11102461236

May, William B./Building emulation comput
QC100 .U56 NO.85-3291 V1985 C.1 NBS-PUB-

3291

Building Emulation Computer Program for Testing of Energy Management and Control System Algorithms

Reference

NBS

PUBLICATIONS

William B. May, Jr.
Cheol Park

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
National Engineering Laboratory
Center for Building Technology
Building Equipment Division
Gaithersburg, MD 20899

December 1985

Sponsored by:
Naval Civil Engineering Laboratory
U.S. Department of Defense

QC100 .U56 NO.85-3291 V1985 C.1 NBS-PUB-
Center of Building and Community Systems
Department of Energy

.U56

85-3291

1985

QC
10
USG
85-3291
1985

NBSIR 85-3291

**BUILDING EMULATION COMPUTER
PROGRAM FOR TESTING OF ENERGY
MANAGEMENT AND CONTROL SYSTEM
ALGORITHMS**

William B. May, Jr.
Cheol Park

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
National Engineering Laboratory
Center for Building Technology
Building Equipment Division
Gaithersburg, MD 20899

December 1985

Sponsored by:
Naval Civil Engineering Laboratory
U.S. Department of Defense

Office of Building and Community Systems
U.S. Department of Energy



U.S. DEPARTMENT OF COMMERCE, Malcolm Baldrige, *Secretary*
NATIONAL BUREAU OF STANDARDS, Ernest Ambler, *Director*

ABSTRACT

A building emulator can be used to test energy management and control systems (EMCS). The emulator uses a computer program to simulate the responses of a building including the equipment, building space, and building envelope to EMCS commands. Building model software for the emulator has been developed at the National Bureau of Standards (NBS) in an effort to assist the United States Naval Civil Engineering Laboratory (NCEL), which is developing a sophisticated building emulator.

The concept of the building emulator and the building emulator computer program are described in this report. The program includes the weather, the air handling unit, the zone, and the comfort model. In addition, the energy compilation routine is also included. The models presented here are simplified models. With these abridged models, a single zone building with exterior walls and a single deck air handling unit are simulated. A complete FORTRAN source code of the building emulator computer program is appended.

KEY WORDS: air handling unit simulation model; building emulator; building space zone model; computer simulation; EMCS algorithms; energy management and control systems (EMCS); HVAC emulator; local equipment simulation model; testing EMCS algorithms.

ACKNOWLEDGEMENT

The authors are indebted to the Naval Civil Engineering Laboratory, Department of Defense, and the Building Systems Division, Department of Energy, for funding this project.

TABLE OF CONTENTS

	<u>Page</u>
ABSTRACT.....	iii
ACKNOWLEDGEMENT.....	iv
LIST OF FIGURES.....	vi
LIST OF TABLES.....	vii
CONVERSION FACTORS FROM METRIC (SI) TO ENGLISH UNITS.....	viii
1. INTRODUCTION	1
1.1 The Concept of a Building Emulator.....	1
1.2 Existing and Planned Building Emulators.....	2
1.3 Testing of EMCS Algorithms with the NCEL/NBS Building Emulator..	2
1.4 Software Components of the NCEL/NBS Building Emulator.....	3
2. BUILDING SIMULATION COMPUTER PROGRAM	5
2.1 Data Structures and Interface to Building Emulator.....	5
2.2 General Program Organization and Control.....	7
2.3 Initialization Routines.....	10
2.4 Simulation Control Routines.....	15
2.5 Building Component Model Routines.....	15
3. WEATHER MODEL	18
3.1 Assumptions and Purpose of Weather Model.....	18
3.2 Outside Air Temperature.....	18
3.3 Solar Energy.....	18
4. AIR HANDLING UNIT MODEL	21
4.1 Assumptions.....	21
4.2 Equations.....	21
5. BUILDING SPACE ZONE MODEL	34
5.1 Assumptions.....	34
5.2 Fundamental Equations.....	35
5.3 Solution of Equations.....	38
5.4 Equations for Heat and Mass Transfer Coefficients and Terms.....	43
5.5 Property Equations for Moist Air.....	53
5.6 Local Equipment Model.....	53
6. OCCUPANT COMFORT MODEL	57
6.1 Assumptions and Purpose of Comfort Model.....	57
6.2 Comfort Envelope Equations.....	57
7. COMPILATION OF ENERGY USE	58
7.1 Categories of Energy Use.....	58
7.2 Equations for Energy Use.....	59
8. SUMMARY.....	62
9. REFERENCES.....	63
APPENDIX. COMPUTER PROGRAM LISTINGS.....	65

LIST OF FIGURES

Figure 2-1. Logic diagram for emulator model program..... 9

Figure 4-1. Block diagram of components of generalized HVAC system
model..... 22

Figure 4-2. Block diagram of components of the abridged, single deck
air handling unit model..... 23

Figure 4-3. Air handling unit model calculation sequence..... 26

LIST OF TABLES

Table 2-1.	Emulator model state vector - Analog sensor variables.....	6
Table 2-2.	Emulator model state variables - Energy variables.....	7
Table 2-3.	Emulator model state variables - Command variables.....	7
Table 2-4.	Emulator model state variables - Comfort variables.....	8
Table 2-5.	Sample simulator configuration file.....	11
Table 2-6.	Sample state vector initialization file.....	12
Table 2-7.	Sample building use parameter file.....	13
Table 2-8.	Sample building shell parameter file.....	14
Table 2-9.	Sample HVAC system parameter file.....	14
Table 2-10.	Sample weather data parameter file.....	16
Table 4-1.	Nomenclature for the air handling unit model.....	28
Table 5-1.	Nomenclature for equations in the building space zone model.....	36
Table 5-2.	General form of the FORTRAN function for a coefficient function.....	43
Table 5-3.	Occupant sensible and moisture gains.....	46
Table 5-4.	Moist air property routines.....	54
Table 7-1.	Heating and cooling energy from emulator model.....	61

CONVERSION FACTORS FROM METRIC (SI) TO ENGLISH UNITS

Physical Characteristic	To Convert from	To	Multiply by
Length	m	ft	3.28
Area	m ²	ft ²	10.76
Velocity	m/s	mph	2.24
Temperature	°C	°F	$t_F = 1.8t_C + 32$
Temperature difference	°C	°F	1.8
Energy	J	Btu	0.948×10^{-3}
Power	W	Btu/hr	3.41
U-value	W/m ² ·°C	Btu/hr·ft ² ·°F	0.176
Thermal resistance	m ² ·°C/W	hr·ft ² ·°F/Btu	5.678
Pressure	kPa	in. Hg	0.296

1. INTRODUCTION

Advances in computer and communications technology have made feasible the use of computers for controlling the heating, ventilation, and air conditioning (HVAC) systems of buildings. Such control systems are usually called energy management and control systems (EMCS). As for any building equipment, it is desirable to have standard testing and rating methods for EMCS. One benefit of such standards is that a more intelligent choice of an EMCS for a building can be made if systems from several manufacturers are tested using standard methods. Once an EMCS is installed, acceptance testing of the system is possible by comparing field test results to design specifications.

1.1 The Concept of a Building Emulator

One approach to testing of EMCS systems is the use of a building emulator. A building emulator is a special computer connected to the sensor inputs and command outputs of an EMCS. No actual HVAC equipment or building is connected to the EMCS. The emulator uses a computer program to simulate the responses of a building (including the equipment, building space, and building envelope) to EMCS commands. The EMCS then controls this simulated building as if it were an actual one. The advantages of using an emulator are that an EMCS may be tested with any type of building if a model is available, and tests can be repeated on different EMCS under the same conditions. Although experience with emulators is limited, it appears that the emulator testing approach is the most desirable in terms of the cost of testing, reproducibility of results, and control of test conditions.

A building emulator can be used to test EMCS algorithms, which may be defined as the software programs in the EMCS which use the EMCS sensors to collect information, make decisions based on current and past information and parameters, and make changes to the building through the EMCS actuators. EMCS algorithms can either be direct digital control (DDC) to control valves and dampers directly, for example, or supervisory to determine start/stop times and setpoints, for example. If a building emulator is used to test EMCS algorithms, it is not necessary to know the exact structure of the algorithm software. Since EMCS software is often proprietary, this is a definite advantage.

Before a building emulator can be used, it must be possible to connect an emulator to an EMCS to be tested. The recommended approach is the connection of the emulator to the EMCS in place of the regular EMCS sensors and actuators. Since it is likely that the EMCS sensors are electrical in nature, they can be replaced with voltage or current sources under the control of the emulator. The EMCS will be unable to distinguish between an actual sensor producing an electrical signal and the emulator producing the same electrical signal. This method allows the EMCS algorithm to be tested directly without the effects of sensors. This implies that a building emulator cannot be used

to test the instrumentation on an EMCS since no sensors or actuators are involved.

1.2 Existing and Planned Building Emulators

Two building emulator prototypes have been built in order to demonstrate the concept of EMCS testing using an emulator. At the United States National Bureau of Standards (NBS), a simple building emulator was constructed in 1983. This emulator demonstrated the basic concept of a building emulator but was not transportable, was limited in capacity, and had a model which was too simple for quantitative algorithm testing.

A second example of a prototype emulator was developed by a contractor for the United States Naval Civil Engineering Laboratory (NCEL) in 1983 [Wise (1984)]. This device, which was referred to as an EMCS exerciser, was designed to be transportable and was used in field tests of actual EMCS installations. Complex software in the exerciser allowed an operator to easily prepare for and run tests. The tests were intended to determine if EMCS supervisory algorithms were operating properly. The exerciser produced reports of the time of occurrence and nature of all EMCS actions during the test. These reports were compared to reports from the EMCS to determine if any problems existed. The unit was not designed to allow a quantitative testing of algorithms, only to determine if the algorithms were operating and controlling the correct command outputs. The exerciser did work as designed and was capable of revealing areas requiring attention in real EMCS installations.

NCEL, with the assistance of NBS, is currently developing a building emulator of greater sophistication than the examples given above. This emulator will have the field transport and general connection features of the NCEL exerciser plus a detailed building simulation model and improved control and evaluation software. NBS is responsible for developing the building model to be used in the NCEL emulator. This report is intended to describe the building model software which has been developed by NBS. The NBS building model does not simulate a complete building, but simulates a single building zone with exterior walls and an air handling unit.

The NCEL/NBS emulator will be used for either factory or field acceptance tests of EMCS algorithms. A factory test will be performed at an EMCS manufacturer facility to determine the capabilities of algorithms on a specific EMCS. A field acceptance test will be used to determine whether an EMCS installed at a specific site is operating in accordance with design specifications. The initial version of the emulator will be designed to test supervisory EMCS algorithms and will not test DDC algorithms.

1.3 Testing of EMCS Algorithms with the NCEL/NBS Building Emulator

Eight types of supervisory control algorithms have been selected for testing

with the NCEL/NBS building emulator. The concepts of the algorithms will not be discussed in this report. Discussion of these EMCS algorithms may be found in other reports [May (1983), Park (1983), Park, Kelly, Kao (1984), Park (1984), and May (1984)]. The types of algorithms to be tested are listed below:

1. Scheduled Start/Stop
2. Duty Cycling
3. Demand Limiting
4. Optimum Start/Stop
5. Day/night Setback
6. Economizer
7. Ventilation/Recirculation
8. Air Setpoint Reset

The emulator model described in this report will not currently support the testing of the demand limiting algorithm. A model of electrical demand in a building would have to be added to the emulator simulation software.

The test procedure to be used for testing EMCS algorithms will determine the exact characteristics that the NCEL building emulator will have. The final form of the test procedure is not known as of the writing of this report. The test procedure cannot be determined entirely by objective means and must result from negotiations between building owners, equipment manufacturers, testing personnel, and standards organizations. For the NCEL emulator it will be assumed that the test procedure will be based on the concept of testing of computers using a 'benchmark' computer program which exercises various operations of any computer on which it is run. Like a program which runs at different speeds depending on the computer on which it is executing, an EMCS algorithm will behave differently depending on the building and climate where it is tested. If 'benchmark' or 'reference' buildings are used for a building emulator, then results of testing different algorithms of the same type on these buildings may be compared.

The building simulation program in the emulator will simulate the response of a building to EMCS control actions. The emulator will be used to provide reproducible predictions of energy use and temperature levels for a set of hypothetical 'reference' buildings which are being controlled by the EMCS system under test.

1.4 Software Components of the NCEL/NBS Building Emulator

The NCEL/NBS emulator computer will employ a multi-tasking disc operating system which will allow several software tasks or processes to effectively appear to be running concurrently. When the emulator is being used to test an EMCS algorithm, there will be two important processes running. One of these is the building simulation program which is described in this report. The other process is an input/output (I/O) program which will read from and write to the

hardware connected to the EMCS sensors and actuators. These two processes will have a common data area in computer memory that both may read from and write to.

If the EMCS sends a command by changing the state or signal level of an EMCS command output, the I/O program detects this and changes the data area that the simulation program uses for command input variables such as start/stop fan, or supply air setpoint. Based on the input variables and model parameters, the simulation program predicts values for variables such as temperatures in the building and HVAC equipment and status points. The simulation program places these values in the common memory area. The I/O program reads these values from the common area and sets the signal levels on the EMCS sensor inputs.

In addition to the model and I/O software, the NCEL emulator will have control software. An operator control program will be used to start and stop the simulation and I/O programs, select parameter files to be used, and determine duration of testing. At the end of tests the simulation program will deposit accumulated energy use by the emulated building into data files. Evaluation software will be used to examine test results and produce evaluations of specific EMCS algorithms such as scheduled start/stop, demand limiting, or setpoint reset.

The next section of this report describes the model software for the emulator in detail. No additional information will be given on the other software components of the emulator since these were not developed at the time this report was written.

2. BUILDING SIMULATION COMPUTER PROGRAM

The building emulator model computer program that was completed by NBS was designed to simulate any type of HVAC system and all possible modes of heat transfer in a zone. However, not all subroutines were completed in order to create a more simplified version. This simplified version will be termed the abridged version. The extent of simplifications will be noted in the sections following.

The building emulator building simulation computer program was written completely in the high level computer language FORTRAN 77 and is appended in the APPENDIX. This section will describe the actual computer program that was developed but will not present the specific equations used to model the building. The emphasis in this chapter will be on describing the data structures, data interface, programming logic and component parts of the program. Following chapters will present detailed equations for the component models of the simulation program.

2.1 Data Structures and Interface to Building Emulator

The building emulator model program is connected to the other programs in the emulator by three data paths. These are the common data area, parameter input files, and report output files. In terms of the building model, the parameter input files control duration of the simulation, timestep, initial conditions, and configure the building systems to be simulated. The report files are the program output, and the common data area represents the state of the simulation and allows input of time dependent boundary conditions.

The common data area is implemented in the emulator model program as a FORTRAN common block. Within the model program, this common block is given the name STATE1 and contains 100 real variables and 100 logical variables. The common block STATE1 contains the 'state vector' for the model. This means that at any time, the values of the variables in STATE1 describe the state of the simulation. Tables 2-1 through 2-4 list all of the variables in the state vector and the engineering units used. The number on the left is the position of the variable in the state vector array. Not all of the state vector positions are used to allow for any future expansion of the model. The variables in table 2-1 are all set by the emulator model program. Selected values in the common area will be converted to electrical signals on the EMCS inputs.

The variables listed in table 2-2 are energy variables which contain cumulative energy used by the building in various categories. This information is written to report files at the end of an emulator test run. More information on energy variables is given in section 7.

Table 2-3 lists variables which the emulator model program uses as input (independent variables). The values of these variables in the common area are set depending on what commands have been received from the EMCS. Some of the variables are logical and some are real variables. The logical variables have two states for use in controlling a two-state device such as a fan. The real variables are used to determine values such as setpoints.

The variables in table 2-4 are set by the emulator program but are not converted to signals for the EMCS. These are comfort variables which are written to a report file at the end of a test period.

Table 2-1. Emulator model state vector - Analog sensor variables

01:	TEMPERATURE of outside air (F)
02:	HUMIDITY RATIO of outside air (lbw/lbda)
03:	ENTHALPY of outside air (Btu/lb)
04:	Windspeed (MPH)
06:	TEMPERATURE of zone air (F)
07:	HUMIDITY RATIO of zone air (lbw/lbda)
08:	TEMPERATURE of zone wall interior (F)
09:	TEMPERATURE of Zone wall surface (F)
10:	TEMPERATURE of zone glass (F)
20:	TEMPERATURE of return air (F)
21:	ENTHALPY of return air (BTU/lb)
22:	HUMIDITY RATIO of return air (lbw/lbda)
23:	TEMPERATURE of mixed air (F)
24:	ENTHALPY of mixed air (BTU/lb)
25:	HUMIDITY RATIO of mixed air (lbw/lbda)
26:	TEMPERATURE of cooling coil discharge air (F)
27:	ENTHALPY of cooling coil discharge air (BTU/lb)
28:	HUMIDITY RATIO of cooling coil discharge air (lbw/lbda)
29:	TEMPERATURE of heating coil discharge air (F)
30:	ENTHALPY of heating coil discharge air (BTU/lb)
31:	HUMIDITY RATIO of heating coil discharge air (lbw/lbda)
32:	TEMPERATURE of supply air (F)
33:	ENTHALPY of supply air (BTU/lb)
34:	HUMIDITY RATIO of supply air (lbw/lbda)
35:	ENTHALPY of return air plus ventilation air (BTU/lb)
51:	MASS FLOW RATE of return air (lb/s)
52:	MASS FLOW RATE of supply air (lb/s)
53:	MASS FLOW RATE of ventilation air (lb/s)
60:	HEATING POWER to heating coil (Btu)
61:	COOLING POWER from cooling coil (Btu)
62:	HEATING POWER to local zone equipment (Btu/s)
63:	COOLING POWER from local zone equipment (Btu/s)

The input parameter files are read by the emulator model program during the initialization of the program. These files are described in section 2.3. The input parameters files are either model parameter files or test control parameter files.

2.2 General Program Organization and Control

The emulator building model computer program is written in the form of a FORTRAN subroutine which can be called from the main program section of the emulator model process running in a multi-tasking system (see section 1.4). The main logic diagram for the emulator model is shown in figure 2-1. It is assumed that the emulator subroutine is called repeatedly without any time delays between calls.

Table 2-2. Emulator model state variables - Energy variables

64: HEATING ENERGY (requirements) (Btu)
65: COOLING ENERGY (requirements) (Btu)
66: HEATING ENERGY (economizer) (Btu)
67: COOLING ENERGY (economizer) (Btu)
68: HEATING ENERGY (load) (Btu)
69: COOLING ENERGY (load) (Btu)
70: HEATING ENERGY (reheat) (Btu)
71: COOLING ENERGY (recool) (Btu)
72: ELECTRICAL ENERGY to fans (kWh)
73: HEATING ENERGY to local zone equipment (Btu)
74: COOLING ENERGY from local zone equipment (Btu)

Table 2-3. Emulator model state variables - Command variables

80: CONTROL POINT ADJUSTMENT supply air temperature (F)
81: CONTROL POINT ADJUSTMENT zone temperature (F)
82: CONTROL POINT ADJUSTMENT outside air damper position (F)
83: CONTROL POINT ADJUSTMENT supply air RH (%)
Logical 01: COMMAND air handling unit on/off
Logical 02: COMMAND economizer on/off
Logical 03: COMMAND ventilation on/off
Logical 04: COMMAND zone setback on/off
Logical 10: STATUS air handling unit on/off
Logical 11: STATUS zone occupied/unoccupied

Table 2-4. Emulator model state variables - Comfort variables

90: COMFORT time outside of comfort range (S)
91: COMFORT maximum dry bulb temperature (F)
92: COMFORT minimum dry bulb temperature (F)
93: COMFORT relative humidity at maximum temperature (%)
94: COMFORT relative humidity at minimum temperature (%)

The emulator program consists of three parts. These are the model equations section, the control section, and the initialization section. During the initialization of the program, a simulation time step, for example thirty seconds, is selected. In a conventional simulation, the control section of the program would advance the variables which contain the time of day and cause the model equations to be solved. The conventional simulation will execute at the full speed of the computer. For an emulator, however, the simulation is artificially slowed so that the time intervals as measured by a real time clock between solutions of the model correspond to the value of the simulation time step. The control software for an emulator must obtain the current time of day from a real-time clock in the computer and only allow the model equations to be solved when a real amount of time has passed.

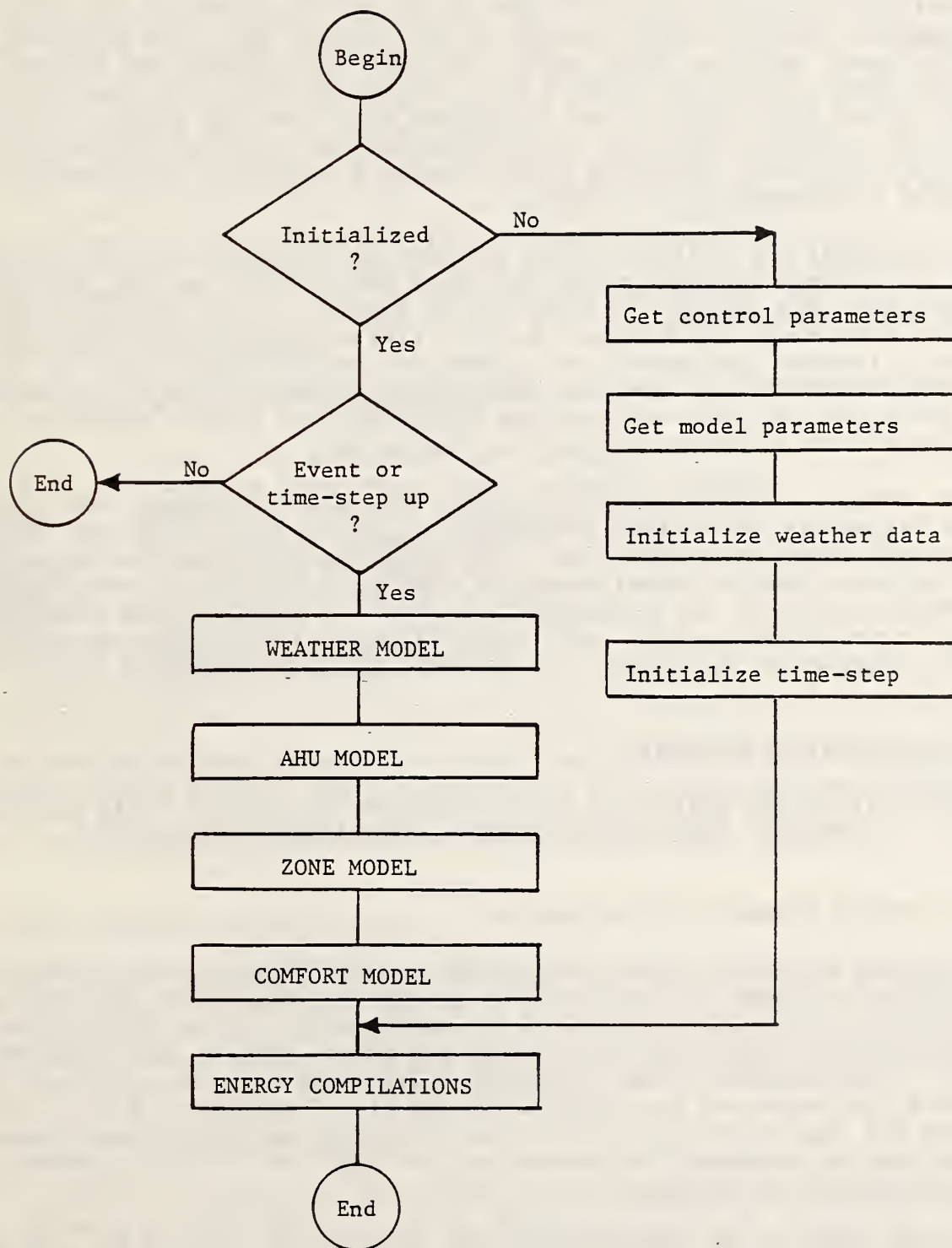


Figure 2-1. Logic diagram of the emulator model program

In addition to monitoring a real-time clock for the passage of the time-step, the emulator building model program control section must also determine if a command event has taken place before the time-step interval has elapsed. If a command event occurs, the program must temporarily adjust the time step to be equal to the time interval that has passed since the most recent solution of the model equations and cause the model equations to be solved immediately. If this is not done, there may be unacceptable delays in the response of the emulator to commands from the EMCS.

Each time that the emulator model subroutine is called, a determination is made as to whether initialization has taken place. If it has not, the program initializes. This process is described in section 2.3. If initialization has already occurred, the program control section determines if either the timestep interval has passed, or a command event has taken place. If neither of these conditions has occurred, then the subroutine is exited. If either of the conditions has occurred, then the model equations section causes the model equations to be solved for the current time of day.

After either initialization or model equations solution, the program calls the energy compilation subroutine. If called after equations solution, the energy compilation subroutine determines the energy used by the emulated building since the last model equations solution and stores a cumulative total of energy used since the beginning of the emulator model program execution. If called after initialization, the cumulative energy totals are reset. After the energy compilation is finished, the emulator model subroutine is exited.

2.3 Initialization Routines

The initialization section of the emulator model program loads simulation control parameters, model parameters, and initializes time variables.

2.3.1 Control Parameter Initialization

The control parameter input subroutine, named CONPAR, is called during the initialization phase to load control parameters. The control parameters are previously stored in a file called the configuration file. CONPAR opens the configuration file and reads the control parameters from it. Table 2-5 shows a sample configuration file. All reading from the file is performed using FORTRAN list-directed read operations and therefore quotes must be placed around the character information, but otherwise no formatting or special delimiters are required. The entries on the right are strictly comments and are not read by the program.

The first entry in the configuration file is the start time of the test using the building emulator. In the example configuration file, this is 12:30. The next entry is the duration of the test in days, hours, minutes, and seconds. In table 2-5, the example test duration is one day and three hours, or a total

of 27 hours. The next four entries in the configuration file are the names of files containing the model parameters. The contents of these files are discussed in the following sections. The names of these files would normally be changed to select a different set of model parameters to allow a different building to be simulated, or the same building to be simulated in a different climate. The seventh entry in the configuration file is the name of the file used to initialize the state vector. The eighth configuration file entry is the weather 'season' to use for the simulation. This will be discussed more fully in section 2.3.2.5. The last entry in the configuration file is the simulation time-step. The time-step is discussed in section 2.2 and section 2.4. The smaller the timestep, the greater the accuracy of the results, but the program execution time rises quickly with decreasing time step. A larger timestep decreases execution time up to a point. Beyond this point the execution time does not change.

Table 2-5. Sample simulation configuration file

12 30 0	'start time: hours, minutes, seconds'
1 3 0 0	'execution time: days, hours, minutes, seconds'
'HVAC.PAR'	'BUILDING HVAC SYSTEM PARAMETER FILE'
'USE.PAR'	'BUILDING USE PARAMETER FILE'
'CLIMATE.PAR'	'WEATHER DATA PARAMETER FILE'
'SHELL.PAR'	'BUILDING SHELL PARAMETER FILE'
'WINTER.INI'	'STATE VECTOR INITIALIZATION FILE'
'WINTER	'SEASON TO USE FOR EMULATION'
20	'Timestep IN SECONDS'

After the control parameters have been read, the CONPAR subroutine calculates the absolute ending time for the emulator test run by adding the test duration to the start time. The ending time is used to determine when to halt the execution of the emulator model program and write the report files.

2.3.2 Model Parameter Initialization

After the control parameters are initialized, the initialization section of the emulator model program calls the subroutine named PARAMA. This subroutine is used to initialize all model parameters. The model parameters are stored in several FORTRAN common blocks which are installed in the PARAMA subroutine and in the various model equation solution subroutines.

2.3.2.1 State Vector Initialization

The first function performed by the PARAMA subroutine is the initialization of the state vector. The variables in the building emulator simulation are located in a large array in a FORTRAN common block. This state vector contains 100 real variables and 100 logical variables. During the solution of the model equations, some of the state variables are calculated completely from the

current values of other variables and do not need to be initialized. Other state variables are calculated from the previous value of the state variable one time step back as well as from the current value of other variables. This latter type of variable must be given an initial value. The state vector variables are initialized using values read from the state vector initialization file. Table 2-6 shows a sample initialization file.

Table 2-6. Sample state vector initialization file

06	52.0	'TEMPERATURE of zone air (F)'
07	0.008	'HUMIDITY RATIO of zone air (lbw/lbda)'
08	21.0	'TEMPERATURE of zone wall interior (F)'
09	48.0	'TEMPERATURE of Zone wall surface (F)'
10	73.0	'TEMPERATURE of zone glass (F)'
80	81.0	'CONTROL POINT ADJUSTMENT supply air temperature (F)'
81	55.0	'CONTROL POINT ADJUSTMENT zone temperature (F)'
83	50.0	'CONTROL POINT ADJUSTMENT supply air RH (%)
92	200.0	'COMFORT minimum dry bulb temperature (F)'
01	.FALSE.	'COMMAND air handling unit on/off'
02	.FALSE.	'COMMAND economizer on/off'
03	.FALSE.	'COMMAND ventilation on/off'
04	.FALSE.	'COMMAND zone setback on/off'
10	.FALSE.	'STATUS air handling unit on/off'

There are three entries on each line of the initialization file. The first entry on the line is the position in the state vector to be initialized. This is a number from 1 to 100. It is assumed that the real numbers are initialized first. The lines in the file must have state vector position numbers in ascending numerical order. When a position number for a line is lower than the number in the preceding line, it is assumed that the following lines will now be used to initialize logical variables.

The second entry on the lines in the initialization file is the initial value to be assigned to the state vector variable. This is a real number constant for the real state variables and a logical constant for the logical state variables. The third entry on the initialization file lines is a comment identifying the name of the state variable. This is not read by the program.

2.3.2.2 Building Use Parameter Initialization

The second function performed by the PARAMA subroutine is the initialization of building use parameters. The building use parameter file contains information on the occupancy and equipment operating times for the building. An example of a building use parameter file is listed in table 2-7.

In the building use parameter file each line is used to determine a change in

the usage status of the building. Between the times indicated by the entries, the usage status is assumed to remain the same. The first column indicates the time in decimal hours that the change in building usage is to occur. The second column indicates the level of lighting in the space in watts. The third column determines the type of lighting used in the building. 'F' indicates fluorescent, and 'I' indicates incandescent. The fourth column is the power used by equipment in the zone in watts. The fifth column is the ratio of radiated energy to total energy for the equipment. The sixth column is the zone moisture gain rate due to the equipment in pounds of water per hour. The last five columns are counts of the number of occupants in one of five activity levels with Activity 5 being the most active. The activity levels are described in section 5.4.5.2.

Table 2-7. Sample building use parameter file

HRS	LIGHT(W)	TYPE	EQUIP,SENS(W)	EQUIP,LAT(LB/H)	ACT 1	ACT2	ACT3	ACT4	ACT5
0.0	100.	'F'	0. 0.5	0.	0	0	0	0	0
8.0	500.	'F'	200. 0.5	0.25	16	8	4	0	0
17.0	100.	'F'	0. 0.5	0.	0	0	0	0	0

After the parameters have been read from the file. The PARAMA subroutine performs some simplifying calculations. The purpose of the calculations is to determine the sensible heat gain rate from equipment, the sensible heat gain rate from people, the moisture gain rate from equipment, and the moisture gain rate from people for each usage state. The equations used are described in section 5.4.5.2. The calculated building use parameters are stored in the common block USE in five arrays. The five arrays contain the above calculated parameters and the time of occurrence for each usage state. A parameter in the program determines the maximum number of usage states.

2.3.2.3 Building Shell Parameter Initialization

After initializing the building use parameters, the PARAMA subroutine initializes the building shell parameters. The building shell parameter file contains parameters which describe the characteristics of the zone, zone walls, and zone windows, for the building to be emulated. Parameters describing the thermal capacity and insulation value of the zone enclosure are in this file. Also in the file are parameters for air infiltration and moisture absorption. An example of the contents of the file is given in table 2-8. The meaning of the entries is assumed to be explained by the comments on each line and will not be discussed here. Only the first value on each line is read by the program. No additional calculations are performed on the building shell parameters in subroutine PARAMA. The parameters are stored in a FORTRAN common block with the name SHELL.

2.3.2.4 HVAC System Parameter Initialization

After initializing the building shell parameters, the PARAMA subroutine initializes the building HVAC system parameters. The HVAC System parameter file contains a description of the heating and cooling equipment available in the building to be emulated. An example of this parameter file is given in table 2-9. The meaning of the entries is assumed to be explained by the comments on each line. Only the first value on each line is read by the program.

Table 2-8. Sample building shell parameter file

1000.	'FT ⁻² '	'Area of the building wall exposed to the outside'
4.05E-4	'BTU/S*F*FT ⁻² '	'inner wall film coeff.(STILL AIR, VERTICAL) R0.68'
1.11E-3	'BTU/S*F*FT ⁻² '	'outer wall film coeff.(7.5 MPH WIND) R0.25'
0.282	'FT'	'thickness of wall outside the mass center'
0.695	'FT'	'thickness of wall inside the mass center'
3.00E-2	'BTU/S*F*FT'	'wall conductivity > the mass center (BRICK)
7.12E-5	'BTU/S*F*FT'	'wall conductivity < the mass center (MASONRY)'
100.0	'BTU/F'	'thermal capacity of zone furnishings (300LB WOOD)'
25000.	'FT ⁻³ '	'volume of air in zone (50x50x10 ft) '
0.19	'BTU/LB*F'	'specfic heat of wall outside the mass center'
130.0	'LBM/FT ⁻³ '	'density of wall outside the mass center'
0.22	'BTU/LB*F'	'specfic heat of wall inside the mass center'
52.8	'LBM/FT ⁻³ '	'density of wall inside the mass center'
0.5	'ACH'	'base air infiltration rate without wind and thermal'
3.0		'furnishings moisture absorbtion mass relative to air'

Table 2-9. Sample HVAC system parameter file

68.75	'ft ⁻³ /sec.'	'Air handling unit fans rated supply air volume'
15.0	'kW'	'power requirement of supply fan at rated volume'
10.0	'kW'	'power requirement of return fan at rated volume'
'REHEAT'		'Local equipment type'
05.0	'Btu/sec.'	'Local equipment capacity, heating'
0.40000	'Btu/sec. F'	'Local equipment controller proportional gain'
0.00010	'Btu/s ² F'	'Local equipment controller integral gain'
60.0	'sec.'	'Local equipment controller sensor time constant'
.FALSE.		'Variable Air Volume System?'
.TRUE.		'Return Air Fan in system?'
0.05	'Btu/sec.'	'Return Air Fan air heating rate'
0.10		'Minimum ventilation air, fraction of supply air mass'
.FALSE.		'Dual Deck System?'
0.06	'Btu/sec.'	'Supply Air Fan air heating rate'
.FALSE.		'Humidity Control?'
10.0	'Btu/sec.'	'Cold Deck Heating Coil Capacity'
15.0	'Btu/sec.'	'Cold Deck Cooling Coil Capacity'
20.	'F'	'Local equipment controller setpoint setback'

2.3.2.5 Climate Parameter Initialization

After initializing the building HVAC system parameters, the PARAMA subroutine initializes the building climate parameters. The building climate parameter file contains the parameters used to simulate the weather to which the emulated building is exposed. The information in the file is for a representative day at five different times of the year for a particular climatic area. These times of the year are termed 'seasons' and correspond to winter, late winter, spring, early summer, and summer. During any particular test, parameters for one season, which is selected by an entry in the emulator model configuration file, will be used to generate a one day variation in weather conditions. This day of weather will be repeated for any other days that the test is run. An example climate file is given in table 2-10. The first line in the file identifies the climate description. This is followed by the five groupings of parameters for five seasons. The season title is the first season information entry. The other entries are explained by the adjacent comments. Additional information about these parameters is given in chapter 3.

2.4 Simulation Control Routines

The simulation control section of the program is found in a subroutine given the name ECTROL. ECTROL has two output arguments, TIMEST and GO. TIMEST is an integer variable which returns the current time step for the simulation to use. GO is a logical variable which is returned as true if the model equations should be solved and false if the equations should not yet be solved.

ECTROL uses two subroutines, TIMINC and NUSTEP, to determine values for variables TIMEST and GO. TIMINC has two logical output arguments, STEPDN and TIMEUP. If a time interval equal to the simulation time step has elapsed since the last solution of the model equations, STEPDN is set true, otherwise it is set false. If the duration of the emulator test specified in the emulator configuration file has elapsed, TIMEUP is set true, otherwise it is set false. If TIMEUP is true, ECTROL calls REPORT, which generates the emulator model report files. EVENT is a global logical variable which is true if the command inputs in the state vector have changed since the last model equations solution, or false otherwise. If either EVENT or STEPDN is true, GO is set to TRUE and ECTROL calls NUSTEP. NUSTEP returns in the variable TIMEST the time elapsed since the last solution of the model equations.

2.5 Building Component Model Routines

The solution of the model equations is effected by successive calls to WEATHR, the weather simulation routine, AHU, the air handling unit simulation routine, ZONE, the zone simulation routine, COMFRT, the comfort model routine, and CMPILE, the energy use compilation routine. All of the models, with the

exception of ZONE, are steady state models not involving differential equations.

Table 2-10. Sample weather data parameter file

'Cold humid winter, hot humid summer, moderate sun'

'WINTER'

35.0 'Maximum temperature, F'
15.0 'Minimum temperature, F'
14.0 'Time of maximum temperature, hrs'
0.003 'Humidity ratio
500. 'Peak solar heat gain on horizontal'
10.0 'Average windspeed, MPH'

'LATE WINTER'

40.0 'Maximum temperature, F'
20.0 'Minimum temperature, F'
14.0 'Time of maximum temperature, hrs'
0.003 'Humidity ratio
1000. 'Peak solar heat gain on horizontal'
10.0 'Average windspeed, MPH'

'SPRING'

60.0 'Maximum temperature, F'
35.0 'Minimum temperature, F'
13.0 'Time of maximum temperature, hrs'
0.004 'Humidity ratio
1700. 'Peak solar heat gain on horizontal'
11.0 'Average windspeed, MPH'

'EARLY SUMMER'

80.0 'Maximum temperature, F'
60.0 'Minimum temperature, F'
14.0 'Time of maximum temperature, hrs'
0.010 'Humidity ratio
2000. 'Peak solar heat gain on horizontal'
9.0 'Average windspeed, MPH'

'SUMMER'

85.0 'Maximum temperature, F'
60.0 'Minimum temperature, F'
14.0 'Time of maximum temperature, hrs'
0.013 'Humidity ratio
1800. 'Peak solar heat gain on horizontal'
8.0 'Average windspeed, MPH'

WEATHR determines the outside air temperature using a sinusoidal approximation. All other values such as humidity ratio and windspeed are

assumed constant. Solar gain is calculated geometrically. Section 3. describes the equations solved in the weather model.

AHU is used to determine the energy required to heat or cool a mix of return and outdoor air to the supply air setpoint. The model has heating coils, cooling coils, and an outside air damper. These devices are assumed to be under perfect sequenced control. There is no local economizer control of the dampers. Section 4 describes the equations for the air handling unit model.

The ZONE subroutine performs a solution of 5 simultaneous equations of which three are differential equations and two are steady-state. These equations describe the dynamics of heat and moisture transfer between perfectly mixed zone air and a two-node wall and the outside environment. A model of local zone heating or cooling equipment under PI control is included in subroutine ZONE. Details of the solution technique and the equations used in subroutine ZONE are discussed in section 5.

The routine COMFRT compares the current zone space conditions to the ASHRAE comfort zone and determines if the zone is comfortable. The comfort model equations are described in section 6.

The subroutine CMPILE, using values from the state vector, accumulates the energy used by the building in various categories. The equations used to calculate energy are described in section 7.

3. WEATHER MODEL

3.1 Assumptions and Purpose of Weather Model

The purpose of the weather model is to provide values for the dry bulb temperature, humidity ratio, windspeed, and solar radiation on a vertical surface for each timestep in the simulation. It is assumed that tests of an EMCS system will require the generation of one to five days of continuous weather data. For such a time period, all of the days in the period will be assumed to have identical weather. The humidity ratio and windspeed will be assumed to remain constant throughout the test period. The dry bulb temperature is assumed to vary smoothly in a sinusoidal manner with a period of one day. Solar radiation is calculated from geometric relationships and statistical values for direct solar radiation.

Model parameters for the weather model were listed in table 2-10. These parameters allow the weather model to be modified to represent different climates and seasons within climates to which a building might be exposed within the constraints listed above. The parameters for humidity ratio and windspeed specify the constant value that these variables will have throughout the day. Other specific parameters will be discussed in the appropriate sections below.

3.2 Outside Air Temperature

The model parameters which determine the variation of outside dry bulb temperature with time for the weather model are the maximum daily temperature, the minimum daily temperature, and the time of occurrence of the maximum temperature. For the generation of a sinusoidal temperature profile the amplitude, period, offset, and phase of the sinusoid must be specified. For temperature variation the period is assumed to be 24 hours. The amplitude is assumed to be half of the difference between the maximum and minimum daily temperatures and the offset is assumed to be the average of maximum and minimum temperatures. The phase can be determined from the time of occurrence of the maximum temperature.

Rather than a mathematical sine function, the emulator model uses a table of 24 discrete points on a curve. The values in the table are normalized to lie between -1 and +1 with a zero offset. Values between the specified points are obtained by interpolation. The interpolated value is then scaled with the amplitude and offset to obtain the temperature.

3.3 Solar Energy

The effects of solar energy on the building are not included in the abridged model that was produced at NBS. The prediction of solar irradiation on the

building could be added to the model using the equations in the following sections.

3.3.1 Solar Radiation Intensity on the Building Wall

The total solar irradiation on the outer surface of the building wall (including windows) is given by:

$$I = I_b \cos\theta + I_s + I_g \quad (3-1)$$

where I : total solar radiation

I_b : direct radiation

I_s : diffusive radiation

I_g : ground reflective radiation (assume $I_g = 0$)

θ : angle of incidence of the sun's rays

Direct and diffusive radiation are evaluated from the equations given by ASHRAE (1981), p. 27.2, and Walton (1983):

$$I_b = C_L A \exp[-B/\sin(\beta)] \quad (3-2)$$

$$I_s = C I_b \quad (3-3)$$

where C_L is a clearness number between 0.75 and 1.0, β is the solar altitude angle, and A , B , and C are constants. Monthly values are shown in Table 1 of ASHRAE (1981), p. 27.2. Walton (1983) provides a detailed procedure for obtaining A , B , and C for a given Julian date if desired.

3.3.2 Sine of Solar Altitude

$$\sin \beta = \sin(d) \sin(L) + \cos(d) \cos(L) \cos(h) \quad (3-4)$$

where d : declination angle (degrees) [ASHRAE (1981), p. 27.2]

L : latitude (degrees)

h : hour angle (degrees)

β : solar altitude angle (degrees)

The hour angle is given by:

$$h = 15 (t_s - 12 + E/60 + TZN) - \lambda \quad (3-5)$$

where t_s : standard time of day (h)

E : equation of time (min) [ASHRAE (1981), p. 27.2]

λ : longitude (degree)

TZN : time zone number [Kusuda (1976), p. 9a]

= 4 for Atlantic 7 for Mountain

5 for Eastern 8 for Pacific

6 for Central

3.3.3 Cosine of Incident Angle

[ASHRAE (1981), p. 27.3]

$$\cos \theta = \cos(\beta) \cos(\phi + \gamma) \sin(\nu) + \sin(\beta) \cos(\nu) \quad (3-6)$$

where θ : angle of incidence of the sun's rays

ϕ : solar azimuth angle from south

γ : angle between the normal to the vertical surface and the south
(facing angle)

ν : tilt angle (the angle between the normal to the surface and the
normal to the ground surface)

$\cos(\beta)$: cosine of solar altitude angle
= $1 - \sin^2 \beta$

(3-7)

3.3.4 Sol-air Temperature

The abridged version of the emulator model program uses dry bulb outside air temperature for heat transfer between the wall and the outdoors. A more complete version would use sol-air temperature as given by the following equation for an opaque wall [Threlkeld (1970), p342]:

$$T_{sol} = T_o + \frac{a I}{U_{ow}} \quad (3-8)$$

where a : solar absorptance of the exposed wall

I : solar radiation intensity (direct + diffuse)

T_o : outside air dry-bulb temperature

U_{ow} : thermal conductance of the wall outside of the mass center

4. AIR HANDLING UNIT MODEL

4.1 Assumptions

The air handling unit model uses steady state air enthalpy calculations to determine the amount of energy required for cooling and heating by the unit. Steady state equations are used since the time step used for the air handling unit model is similar in size to the time constants of HVAC systems. Possible EMCS control actions that provide input to the air handling unit model are starting and stopping the air handling unit, closing or allowing modulation of the outside air dampers, opening or closing minimum outside air dampers, and supply air temperature setpoint. The general model is designed to be configurable to represent either a constant or variable air volume (VAV) unit with either single or dual decks and heat exchangers for heating or cooling. The abridged model (section 2.0) that was implemented in the building emulator model computer program does not allow simulation of dual deck systems or systems with humidity control. See subroutine AHU in APPENDIX.

4.2 Equations

A sample of the model parameters for the air handling unit simulation subroutine was listed in table 2-9. The parameters in the table fall into two categories, which are configuration parameters and analog parameters. The configuration parameters are logical constants which are used to select a particular configuration for the air handling unit. There are four of these parameters, which are DUAL-DECK (DUDECK), HUMIDITY-CONTROL (HUMCON), RETURN-AIR-FAN (RAFAN), and VARIABLE-AIR-VOLUME (VAV), and which may be either true or false.

Figure 2-1 is a diagram of a generalized air handling unit showing the major components of the model. The two basic configurations for the air handling unit are dual-deck or single deck. To select dual deck, the logical parameter DUDECK is set true. If dual deck, there will be a set of heat exchangers for two paths and the supply fan will be located before the split. (Dual deck is not implemented in the abridged model) If the system is single deck, DUDECK will be false, the second set of heat exchangers will not be used, and the supply fan will be located after the last exchanger.

The decks in the model may contain a humidifier, a cooling coil, a heating coil or any combination of the three. The return air fan is optional and the logical parameter RAFAN determines whether it will be included in the model. If the parameter VAV is true, the air handling unit will be modeled as being Variable Air Volume (VAV), and the mass flow rate of air through the unit will be allowed to change to match the zone loading. If humidity control is to be used, parameter HUMCON must be set to true, and the supply air will be maintained at a setpoint for humidity as well as a dry bulb temperature setpoint (humidity control is not implemented in the abridged model).

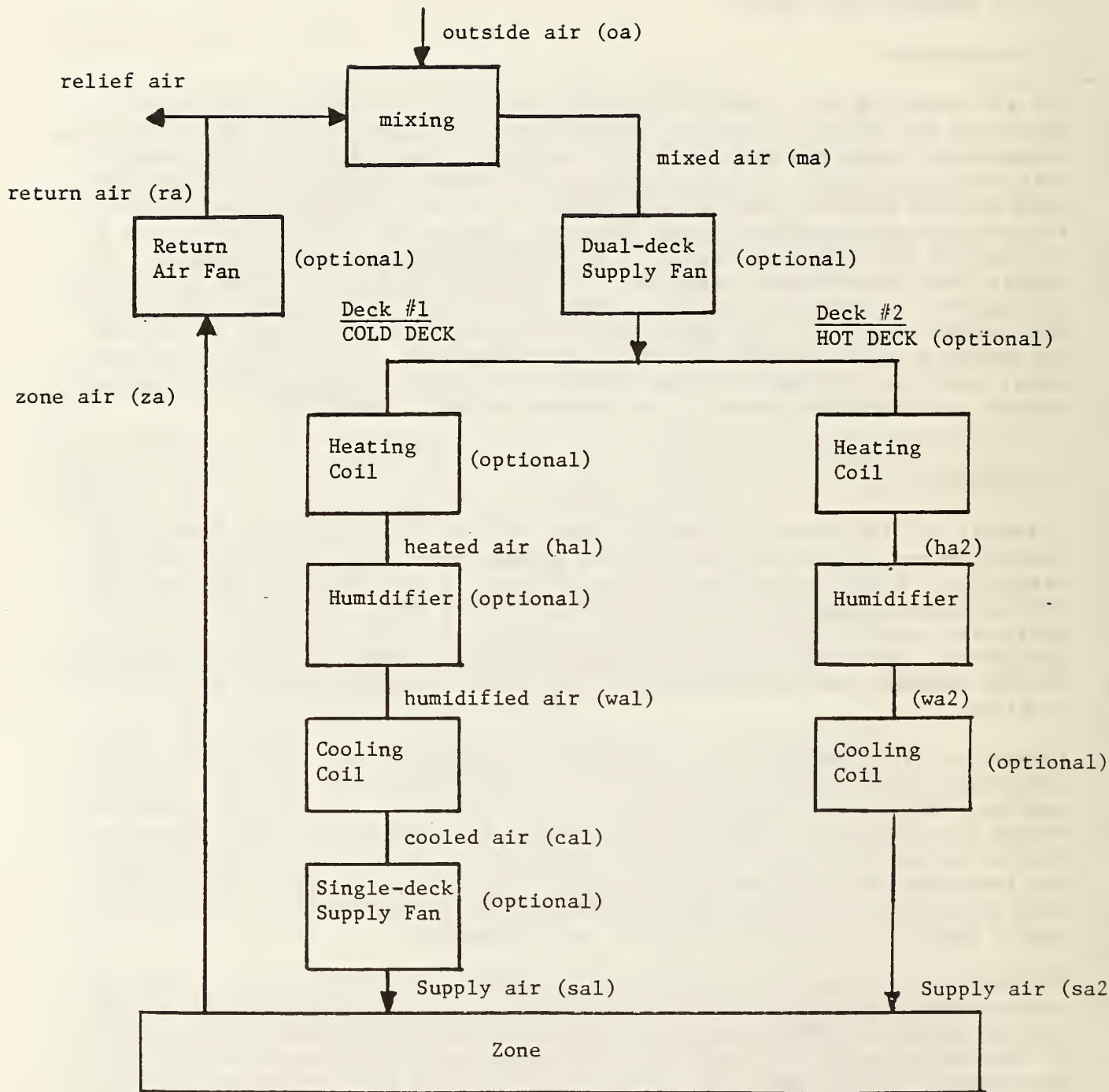


Figure 4-1. Block diagram of components of generalized HVAC system model

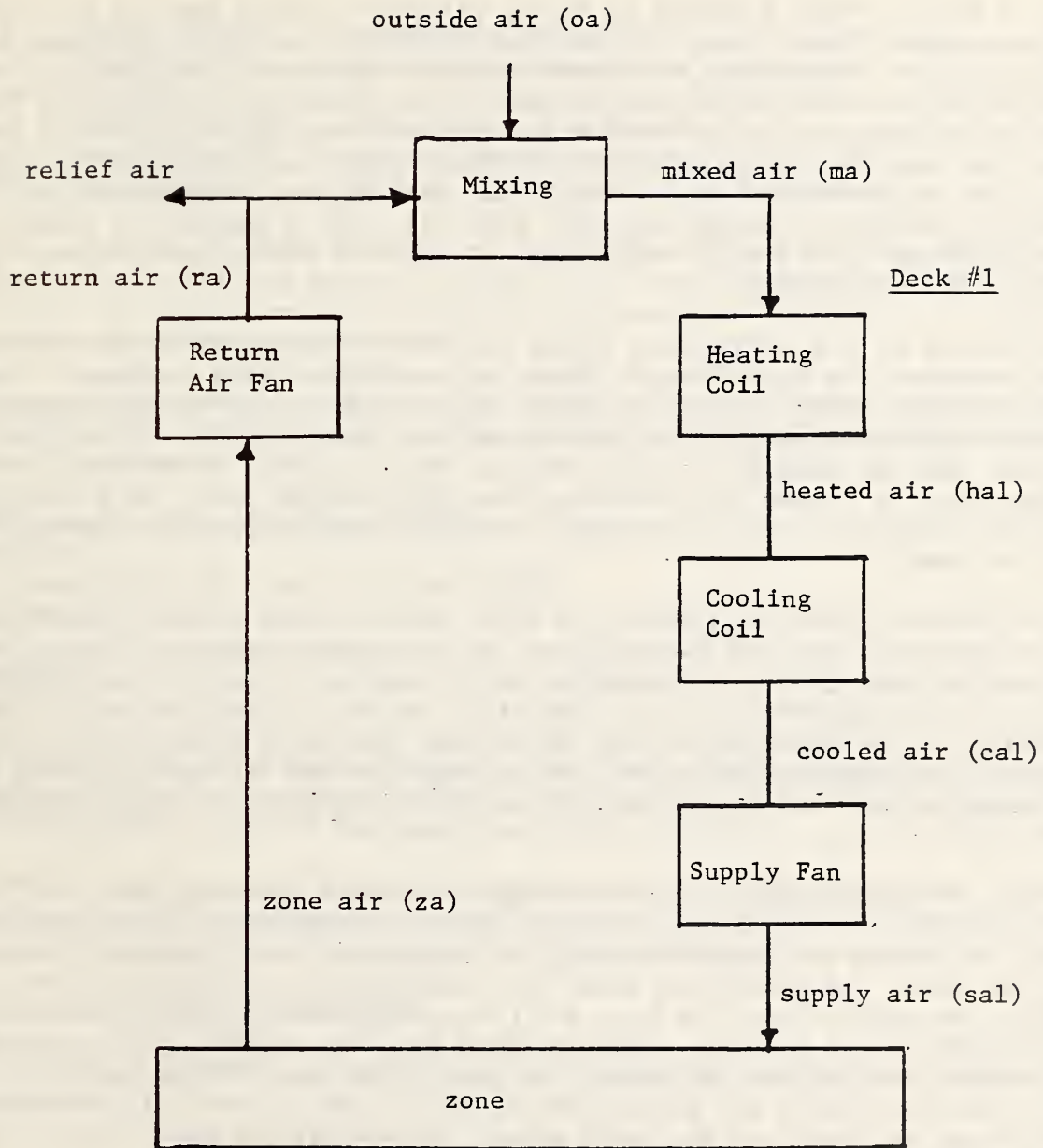


Figure 4-2. Block diagram of components of the abridged, single deck air handling unit model

Table 2-9 also shows a number of analog parameters which have to be specified for the model. These describe the capacities of the coils and fans in the unit. The fans in the unit will have a power requirement, a rated air volume flow rate, and a parameter to specify how much energy was added to the air passing through the fan by the fan waste heat. The fans for VAV will use the rated flow rate as the maximum but will have a flow rate and power consumption which can be lowered to match zone loads. Cooling and heating coils will have a certain capacity in energy per unit time that can be removed or added to the air in the unit. If a coil is given a capacity of zero, this indicates that the coil is not present.

In addition to the parameters there are inputs in the FORTRAN common area which contains the state vector. These are variables which represent commands from the EMCS which is being tested by the emulator. The variable ECONOMIZER (ECONO) will be a signal to use or not use the outside air dampers for cooling. UNIT-ON (UNITON) will be true if the unit is to be running and false if the unit is turned off. VENTILATION-OFF (VENOFF) will be false during occupied operation, but if the EMCS closes minimum outside air dampers VENOFF will be true.

The locations shown in figure 4-1 with two or three letter abbreviations beside them and location descriptions in parenthesis represent states of the air passing through the air handling unit. Each state can be characterized by the dry bulb temperature, T , the humidity ratio, W , the flow rate m and the enthalpy, h . The abbreviations on the diagram represent subscripts to variable names in the equations which are used to describe the air handling unit model. For example, the temperature of the air returning from the building zones is symbolized by T_{za} .

It will be assumed that if the capacity of the air handling unit will allow, the condition of supply air from the the air handler will be the same as specified by the air handler supply air setpoints. For a constant volume air handling unit, the supply air mass flow rate will be constant. If the system is VAV, the supply air flow rate will be determined by the local equipment model contained in the zone model (section 5.6). The return air flow rate will be assumed to equal the VAV supply air mass flow rate. If the system is dual-deck, the individual air mass flow rates for the hot and cold deck will be determined by the local equipment model. The return air mass flow rate will then equal the sum of the two deck mass flow rates.

The following FORTRAN functions are assumed for psychrometric calculations (the implementations of these functions are described in section 5.5). For the arguments listed, T is dry bulb temperature, W is humidity ratio, P is atmospheric pressure, RH is relative humidity, h is enthalpy, and DP is dewpoint temperature (also see Table 5-3).

1. $h = \text{ENTHAL}(T,W,P)$ - enthalpy of moist air
2. $T = \text{TEMP}(h,W,P)$ - dry bulb temperature of moist air
3. $DP = \text{DEWPT}(W,P)$ - dew point temperature of moist air
4. $W = \text{HUMRAT}(T,RH,P)$ - humidity ratio of moist air
5. $RH = \text{RELHUM}(T,W,P)$ - relative humidity of moist air
6. $T = \text{TDBSAT}(h,P)$ - dry bulb temperature of saturated air

Figure 4-2 is a flow diagram showing the logic which determines the equations to use for calculating the states of the air in the air handling unit. The calculations for each time step are performed based on the return air state which is known from the zone air model.

Since the equations used for the air handling unit model depend on the state of several logical variables, a special format will be used to present the equations. The construction:

```

IF (condition1) is true (or false) THEN
    equation 1
ELSE IF (condition2) is true (or false) THEN
    equation 2
ELSE
    equation 3
ENDIF

```

will be used to denote logical decisions about which equation to use. Indentations will be used for clarity. See Figure 4-2 for logic flow and Table 4-1 for nomenclature of the equations which follow.

The following equations will apply if logical variable UNIT-ON is true. If it is not, then the air handling unit is off, no air is flowing, all mass flow rates will be zero, and all energy use by the air handler will be zero.

```

IF Variable-Air-Volume is true THEN

```

The supply air volume is assumed to be determined by the local zone VAV equipment.

```

ELSE

```

Constant volume is assumed. The mass flow rate m_{sa} equals the rated air flow rate.

```

ENDIF (if Variable-Air-Volume is true).

```

```

IF Return-Air-FAN is true THEN

```

Assume that the return air fan causes a constant amount of thermal energy to be added to the return air per second. The return air fan heating rate must be provided as a parameter, q_{rf} . Then the temperature rise due to the return air fan is:

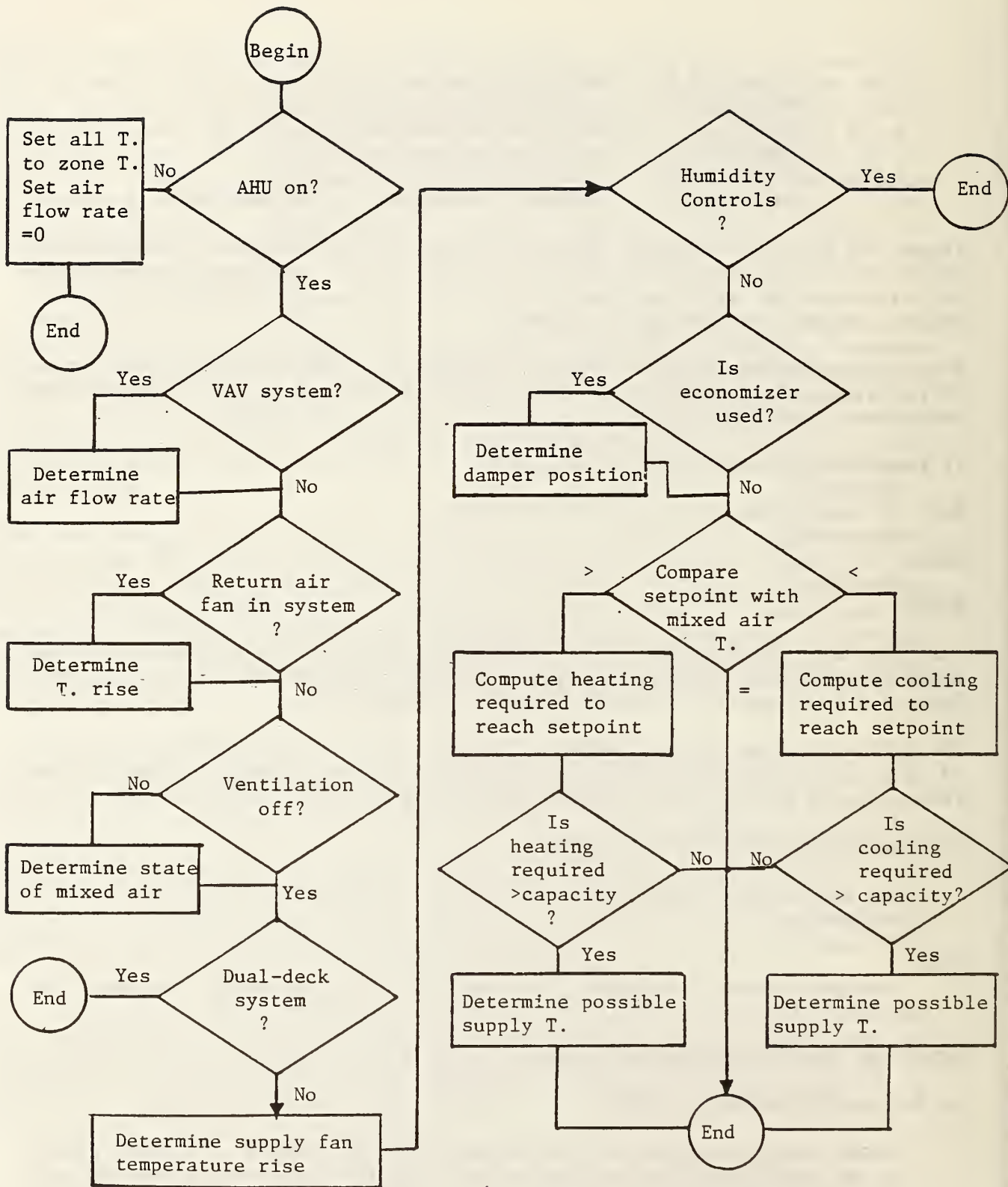


Figure 4-3. Air handling unit model calculation sequence

$$dT_{rf} = \frac{q_{rf}}{m_{ra} c_{pa}} \quad (4-1)$$

$$\text{and } T_{ra} = T_{za} + dT_{rf} \quad (4-2)$$

ELSE

$$T_{ra} = T_{za} \quad (4-3)$$

ENDIF (if Return-Air-FAN is true)

$$W_{ra} = W_{za} \quad (4-4)$$

$$h_{ra} = \text{ENTHAL}(T_{ra}, W_{ra}, P) \quad (4-5)$$

The next calculations determine the state of the mixed air (subscript mc) if the outside air dampers are closed. It will be assumed that f_{va} is a model parameter which the fraction of air mass flow through the air handling unit which is the minimum outside air flow rate for ventilation purposes. If the EMCS cuts off ventilation air as an algorithm action, then this value will be zero:

IF VENTilation-OFF is true THEN

$$h_{mc} = h_{ra} \quad (4-6)$$

$$W_{mc} = W_{ra} \quad (4-7)$$

$$T_{mc} = T_{ra} \quad (4-8)$$

ELSE

$$h_{mc} = f_{va} h_{oa} + (1 - f_{va}) h_{ra} \quad (4-9)$$

$$W_{mc} = f_{va} W_{oa} + (1 - f_{va}) W_{ra} \quad (4-10)$$

$$T_{mc} = \text{TEMP}(h_{mc}, W_{mc}, P) \quad (4-11)$$

ENDIF (if VENTilation-OFF is true)

IF DUal-DECK is false THEN

For a single deck air handling unit, the supply fan temperature rise is calculated as the return fan rise was calculated:

Table 4-1. Nomenclature for the air handling unit model

c_{pa}	specific heat of air
DP	dew point
dT	temperature rise
f_{va}	fraction of air mass flow for the minimum outside air for ventilation
h	enthalpy
m	air mass flow rate
P	barometric pressure
q	heating/cooling rate
$q_{c,max}$	cooling capacity
$q_{h,max}$	heating capacity
RH	relative humidity
T	air temperature
T_{set}	setpoint temperature
W	humidity ratio

Subscripts:

c	cooling
ca	cooled air
h	heating
ha	heated air
ma	mixed air
mc	state of mixed air if the outside air dampers are closed
mo	state of mixed air if the outside air dampers are completely open
oa	outside air
ra	return air
rf	return air fan
sa	supply air
sf	supply air fan
wa	humidified air
za	zone air

$$dT_{sf} = \frac{q_{sf}}{m_{sal} c_{pa}} \quad (4-12)$$

IF HUMidity-CONTROL is false THEN

 If there is no humidity control, there will only be a dry bulb temperature setpoint. The humidity will be allowed to float to any level.

IF ECONOmizer is true THEN

The EMCS allows the air handling unit to use the outside air for conditioning supply air. No provision is made in this version of the model for direct EMCS control of damper position. It is assumed that local controls determine proper outside air damper position. However, once the outside air damper is fully open when cooling is required, it remains open even if outside air enthalpy is high. The EMCS must take responsibility for closing the damper in this case. To determine the outside air damper position:

$$h_{mo} = h_{oa} \quad (4-13)$$

$$T_{mo} = T_{oa} \quad \text{and} \quad (4-14)$$

$$W_{mo} = W_{oa} \quad (4-15)$$

where the mo subscript indicates the state of the mixed air if the outside air dampers are completely open.

If $T_{mc} \leq T_{set} - dT_{sf}$, this indicates that the outside air damper should be closed and that heating of the supply air will be required. In this case:

$$T_{ma} = T_{mc} \quad (4-16)$$

$$W_{ma} = W_{mc} \quad \text{and} \quad (4-17)$$

$$h_{ma} = h_{mc} \quad (4-18)$$

However, if $T_{mc} > T_{set} - dT_{sf}$ and $T_{mo} > T_{set} - dT_{sf}$, this indicates that the outside air damper should be wide open, but additional cooling of supply air will be needed using a cooling coil. In this case:

$$T_{ma} = T_{mo} \quad (4-19)$$

$$W_{ma} = W_{mo} \quad \text{and} \quad (4-20)$$

$$h_{ma} = h_{mo} \quad (4-21)$$

But, if $T_{mc} > T_{set} - dT_{sf}$ and $T_{mo} \leq T_{set} - dT_{sf}$, then the outside air dampers will be able to provide all conditioning of the air passing through the air handler. In this case, the outside air dampers will be at some intermediate position. The supply air temperature will equal the setpoint but since the fraction of outside air is not known in this case, the exact psychrometric state of the supply air (and thus the enthalpy) is not known. Theoretically, since the dry-bulb temperature lines on the psychrometric chart are not parallel, an iterative

technique would be required to determine W_{ma} and h_{ma} . However, the lines are approximately parallel and an approximate relationship can be used since the state of the mixed air lies on a straight line between the state of the air with the dampers fully open and the state with the dampers closed. This is described by

$$\frac{T_{ma} - T_{mo}}{T_{mc} - T_{mo}} = \frac{W_{ma} - W_{mo}}{W_{mc} - W_{mo}} \quad \text{or} \quad (4-22)$$

$$\text{Then, since: } T_{ma} = T_{set} - dT_{sf} \quad , \quad (4-23)$$

$$W_{ma} = \frac{W_{mc} - W_{mo}}{T_{mc} - T_{mo}} (T_{set} - dT_{sf} - T_{mo}) + W_{mo} \quad (4-24)$$

$$h_{ma} = \text{ENTHAL}(T_{ma}, W_{ma}, P) \quad (4-25)$$

ELSE

If the EMCS keeps the outside air dampers closed:

$$T_{ma} = T_{mc} \quad , \quad (4-26)$$

$$W_{ma} = W_{mc} \quad \text{and} \quad (4-27)$$

$$h_{ma} = h_{mc} \quad . \quad (4-28)$$

ENDIF (ECONOmizer is true)

If outside air is providing all of the conditioning, as determined above ($T_{ma} = T_{set} - dT_{sf}$), then no heating or cooling of the mixed air is required, and:

$$T_{cal} = T_{ma} \quad , \quad (4-29)$$

$$W_{cal} = W_{ma} \quad , \quad (4-30)$$

$$h_{cal} = h_{ma} \quad , \quad (4-31)$$

$$T_{hal} = T_{ma} \quad , \quad (4-32)$$

$$W_{hal} = W_{ma} \quad , \quad \text{and} \quad (4-33)$$

$$h_{hal} = h_{ma} \quad . \quad (4-34)$$

However, if heating of the mixed air is required ($T_{ma} < T_{set} - dT_{sf}$) then the air must be heated from T_{ma} to T_{hal} , where:

$$T_{hal} = T_{set} - dT_{sf}. \quad (4-35)$$

The simplified psychrometric process for air heating is to increase the temperature of the air at constant humidity ratio (no moisture is added). Therefore:

$$W_{hal} = W_{ma} \quad \text{and} \quad (4-36)$$

$$h_{hal} = \text{ENTHAL}(T_{hal}, W_{ma}, P). \quad (4-37)$$

Then the energy added to the air is:

$$q_h = m_{sal} (h_{hal} - h_{ma}). \quad (4-38)$$

If q_h is greater than the heating capacity of the air handling unit, then the supply air temperature will be below the supply air setpoint. The supply air temperature is calculated as follows:

$$h_{hal} = \frac{q_{h,max}}{m_{sal}} + h_{ma}, \quad \text{and} \quad (4-39)$$

$$T_{hal} = T(h_{hal}, W_{ma}, P). \quad (4-40)$$

Then the energy added to the air is:

$$q_h = q_{h,max} \quad (4-41)$$

Alternatively, cooling of the air may be required ($T_{ma} > T_{set} - dT_{sf}$). Then the air must be cooled from T_{ma} to T_{cal} , where:

$$T_{cal} = T_{set} - dT_{sf}. \quad (4-42)$$

The assumed psychrometric process for air cooling is to remove energy from the air at constant humidity ratio until the dew point temperature of the mixed air is reached, at which point energy and water are removed from the air, which remains saturated, until T_{cal} is reached.

The dewpoint temperature must be known for the calculations:

$$DP_{ma} = \text{DEWPT}(W_{ma}, P) \quad (4-43)$$

If $T_{cal} > DP_{ma}$, then the supply air will not be at a saturated condition after it is cooled and:

$$h_{cal} = h(T_{cal}, W_{ma}, P) , \text{ and} \quad (4-44)$$

$$W_{cal} = W_{ma} . \quad (4-45)$$

However, if $T_{cal} \leq DP_{ma}$ then condensation of moisture from the air will occur and the supply air will be assumed to be saturated:

$$RH_{cal} = 100 \% , \quad (4-46)$$

$$W_{cal} = HUMRAT(T_{cal}, RH_{cal}, P) , \quad \text{and} \quad (4-47)$$

$$h_{cal} = ENTHAL(T_{cal}, W_{cal}, P) . \quad (4-48)$$

Then the cooling energy required to maintain the setpoint is:

$$q_c = m_{sal} (h_{ma} - h_{cal}) \quad (4-49)$$

If q_c is larger than the rated capacity of the cooling coil, then the supply air state will be above the setpoint of the air handling unit. The supply air state and cooling energy will then have to be calculated as follows:

$$h_{cal} = h_{ma} - \frac{q_{c,max}}{m_{sal}} , \text{ and} \quad (4-50)$$

$$q_c = q_{c,max} \quad (4-51)$$

If $h_{cal} > ENTHAL(DP_{ma}, W_{ma}, P)$, then the supply air at maximum cooling capacity is not saturated:

$$T_{cal} = TEMP(h_{cal}, W_{ma}, P), \text{ and} \quad (4-52)$$

$$W_{cal} = W_{ma} . \quad (4-53)$$

However, if $h_{cal} \leq h(DP_{ma}, W_{ma}, P)$, then the supply air at maximum cooling capacity is saturated:

$$T_{cal} = TDBSAT(h_{cal}, P), \text{ and} \quad (4-54)$$

$$W_{cal} = HUMRAT(T_{cal}, RH=100\%, P) \quad (4-55)$$

The final supply air state is then:

$$W_{sal} = W_{cal} \quad (4-56)$$

$$T_{sal} = T_{cal} + dT_{sf} \quad (4-57)$$

$$h_{sal} = \text{ENTHAL}(T_{sal}, W_{sal}, P)$$

(4-58)

ELSE

If it is in the heating season and the humidity is too low, a humidifier can be used to increase the humidity ratio of the supply air. If it is in the cooling season and humidity is excessive, the air can be dehumidified by overcooling and reheating of air. Humidity control was not implemented in the abridged version of the emulator model and equations will not be presented.

ENDIF (if HUMidity-CONTROL is false)

ELSE

Equations for dual deck were not implemented in the abridged version of the emulator model and will not be presented.

ENDIF (if DUal-DECK is false)

The states of the air passing through the air handling unit are stored in the state vector and are made available to other model subroutines in the building emulator model program.

5. BUILDING SPACE ZONE MODEL

The building zone model subroutine, unlike the other model equation subroutines, implements a model formulated in terms of differential equations. A dynamic rather than a steady-state model is necessary since the time step magnitude used in the model (smaller than one minute) is much smaller than the characteristic time constants of the zone.

5.1 Assumptions

A number of simplifying assumptions are made in order to allow a relatively simple set of model equations. The building zone to be modeled is assumed to be a typical perimeter zone of a large multi-story building as Kao (1983) used in his energy conservation study for a large office building. The zone is enclosed by an exposed wall with a glass window, three interior walls, ceiling, and floor. Since there is one exposed wall and one glass window the governing equations are simplified. The following assumptions are made for the building zone model:

- (1) Uniform temperature distributions on surfaces.
- (2) No stratification of zone air (fully mixed air).
- (3) No heat or mass flow between zones.
- (4) Simple, uniform optical and thermal properties for any glass windows.
- (5) The term 'wall' includes walls, ceiling, and floor.
- (6) All wall structures (including ceiling and floor) which are not exposed directly to the outside are treated as thermally adiabatic. (No conductive heat flow across the wall structure is assumed).
- (7) Simple two-node representation of the exterior wall with one node at the interior wall surface and one node at the mass center point of the wall.
- (8) Convective and radiative heat transfer coefficients for the inner surfaces of walls are considered constant.
- (9) The surface heat transfer coefficient of outside wall surfaces is considered constant and is not a function of outdoor conditions (including wind).
- (10) The wind speed is considered constant and has an influence only on infiltration.

- (11) All static pressures in the HVAC system, zone and outdoor air are assumed to be constant with a value of 29.92 inches of mercury.
- (12) The latent heat gain from furnishings in the zone is neglected.
- (13) The heat capacity of zone furnishings is assumed to remain the same at all times.
- (14) Radiant heat from lights (short wave radiation) is assumed to be uniformly distributed on all inner surfaces of the zone.
- (15) For all glass windows, the same constant values of transmittance and shading coefficient are used.
- (16) For all exposed opaque walls, solar absorptance is a constant.
- (17) Solar radiation intensity consists of direct and diffuse radiation. Ground reflective radiation is ignored. The solar constants A, B, and C appearing in equations (3-2) and (3-3) are assumed constant for a given specific month.
- (18) It is assumed that solar radiation entering the zone through windows is not absorbed by glass but is distributed uniformly on the inner surfaces of the zone.

Based on these assumptions, it is assumed that five state variables are sufficient to model the zone. The state variables are room air dry bulb temperature, room humidity ratio, exterior wall interior surface temperature, glass interior surface temperature, and exterior wall internal temperature.

The abridged model (see section 2) implemented in the building emulation simulation program included some additional assumptions. The equations presented in the following sections will usually be for a complete model. The changes to equations for the abridged model will be described whenever necessary in the following sections. The assumptions for the abridged model are:

- 1) No effects of solar energy.
- 2) No radiative heat transfer between zone surfaces.

5.2 Fundamental Equations

Since there are five state variables in the building emulator, five equations relating to these variables are required to solve for the values of the variables. Table 5-1 contains nomenclature for the fundamental equations of the zone model.

5.2.1 Zone Air Heat Balance

The first fundamental equation for the emulator zone model expresses the balance in heat transfer between the air and furnishings in the zone and the wall and window surfaces, entering supply air from the air handling unit, infiltration heat transfer, and internal sources of heat [Borresen (1981), Modest and Macken (1977), Bullock (1984), Park (1981)]. The equation is:

$$C_{\text{zone}} \frac{dT_i}{dt} = Q_{\text{supply}} + Q_{\text{infiltration}} + Q_{\text{convection}} + Q_{\text{internal}} \quad (5-1)$$

The left hand side of the equation represents the storage of energy in the zone air and furnishings. The state variable is the interior air temperature. Of the four terms on the right side of the equation, the energy from the supply air and the internal gains may be considered boundary conditions. The supply air condition is determined by the air handling unit model and the internal gain, consisting of heat gains from occupants, lights, and equipment, is determined by the building schedule. The other two terms for convection and infiltration heat transfer are functions of the other state variables.

Table 5-1. Nomenclature for equations in the building space zone model

State Variables:

T_i	- space dry bulb temperature	W_i	- space humidity ratio
T_w	- wall mass center temperature	T_{ig}	- window glass temperature
T_{iw}	- wall interior surface temperature		

Other Parameters:

A	- conduction coefficient	B	- convection coefficient
C	- thermal mass	T	- temperature
M_{air}	- air mass for water absorption	Q	- heat transfer rate
T_{mr}	- space mean radiant temperature	R	- radiative coefficient
G	- zone moisture increase rate		
S_{gw}	- wall heat gain by solar radiation through windows		

Subscripts:

convec	- convection	o	- outside air	g	- glass
infiltration	- infiltration	w	- at the wall mass center		

5.2.2 Exterior Wall Mass Heat Balance

The exterior wall is assumed to be the only surface enclosing the zone which exchanges energy with the outdoor environment and the indoor environment. The state variable for wall temperature is assumed to be the temperature at the

mass center of the wall. The mass center is defined as that point in the wall where for a unit area of wall the total mass of the wall outside of the mass center is equal to the total mass of the wall inside the mass center. For a masonry construction, this may result in the mass center being different from the geometric center of the wall. The equation for the exterior wall is:

$$C_{\text{wall}} \frac{dT_w}{dt} = Q_{\text{inside}} + Q_{\text{outside}} + Q_{\text{solar}} \quad (5-2)$$

The left hand side of the equation represents the storage of energy in the building wall. The first term on the right is the heat transfer between the wall interior and the inner wall surface. This term is dependent on state variables. The second two terms on the right represent the heat transfer between the wall interior and the outdoor environment, including the impingement of solar irradiation on the outside of the wall. The right two terms are time-dependent boundary conditions.

5.2.3 Zone Air Moisture Balance

The air in the zone is assumed to contain moisture. The state variable used to describe the water content of the air is the humidity ratio. The equation used to describe the moisture balance is [Chi (1982), Bullock (1984)]:

$$M_{\text{air}} \frac{dW_i}{dt} = G_{\text{infiltr}} + G_{\text{supply}} + G_{\text{internal}} \quad (5-3)$$

The left hand side of this equation represents the storage of water vapor in the air and furnishings of the zone. The right hand side of the equation has three terms. The first term is the moisture transfer with the outdoor environment due to infiltration, which contains a state variable. The second term is the net moisture added to or removed from the room by the air handling unit. This is the difference in absolute moisture between the supply and return air from the air handling unit, and is a boundary condition. The third term is the moisture generated internally by occupants and equipment. The deficiency in this equation is that it assumes the air and furnishings can absorb an infinite amount of moisture when in reality, the air can become saturated. The emulator model zone subroutine performs a check of the relative humidity of the air to prevent oversaturation.

5.2.4 Inner Wall Surface Heat Balance

For the inner wall surface of the exterior wall, the heat balance is expressed by an algebraic equation rather than a differential equation since there is no storage of energy at the wall surface. The equation is:

$$A_w(T_w - T_{iw}) + B_w(T_i - T_{iw}) + R_w(T_{mr} - T_{iw}) + S_{gw} = 0 \quad (5-4)$$

This equation is based on the assumption that all heat transfer in and out of the node at the wall surface sums to zero. The first term is the conduction heat transfer between the wall interior and the inner wall surface. A_w is a conduction coefficient. The second term is the convection heat transfer between the wall surface and the zone mean radiant temperature. The final term is due to any solar irradiated energy which passes through the zone window and is absorbed by the zone inside surfaces.

5.2.5 Inner Glass Surface Heat Balance

For the inner glass surface of the zone window, if present, an algebraic equation describes the heat transfer balance since there is no storage of energy:

$$A_g(T_o - T_{ig}) + B_g(T_i - T_{ig}) + R_g(T_{mr} - T_{ig}) = 0 \quad (5-5)$$

This equation expresses that the sum of all heat transfer in and out of the node at the glass surface is zero. The first term is the conduction/convection heat transfer between the inner glass surface and the outdoor air. The coefficient A_g is a combination convection and conduction coefficient. The second term in the equation is the convection heat transfer between the glass surface and the interior air. The coefficient B_g is for convection. Radiant heat transfer between the glass surface and the mean radiant temperature of the room is represented by the third term of the equation.

5.3 Solution of Equations

The equations described in section 5.2 are solved by the zone model subroutine using a simple explicit forward difference numerical technique (Euler method) to predict the values of the state variables one step forward in time. The time step used within the zone model is different from the emulator time step used for the other models in the emulator simulation program. Since simple differential equation solution technique is used within the zone model, the time steps must be small or the solution will not converge. The emulator time step is divided into a number of smaller zone model time steps and the differential equations are solved for each zone model time step until the

emulator time step has passed. However, the zone model is not constrained to execute zone model time steps in real time.

The subroutine ZONE performs the steps required to determine the values of the state variables at the next emulator time step. This routine first determines the best value for the zone model time step. This process is described in section 5.3.2. Then ZONE calls the two subroutines EULER and SURFAC in sequence, and advances the time by one zone model time step until the emulator time step has passed. The subroutine EULER is used to solve the three differential equations listed above (5-1 through 5-3) for one zone model time step. SURFAC determines the wall and glass surface temperatures for the current time step using the algebraic equations described above (5-4 through 5-5).

5.3.1 Euler Forward Difference Technique

The subroutine EULER is a generalized routine to advance a system of ordinary first order differential equations by one time step using the explicit difference method. The arguments to the routine include the number of differential equations, an array containing the dependent (state) variables, the time step, and the name of a FORTRAN function which will evaluate the right hand sides of the differential equations when they are expressed in the form:

$$\frac{dY_i}{dt} = f(t, Y_1, Y_2, \dots, Y_n), \quad i=1,2,\dots,n \quad (5-6)$$

where Y_i is the i th dependent variable and n is the number of equations. The value of Y at time $t + \Delta t$ can approximately be predicted by:

$$Y_i(t + \Delta t) = Y_i(t) + \Delta t * \text{FUNCTION}(i,t,n,Y_1,Y_2,\dots,Y_n) \quad (5-7)$$

As long as the product of the time step Δt and the function value is small, a solution to the differential equations can be obtained.

5.3.2 Adaptive Time Step

The time step used by the EULER subroutine is determined adaptively to become smaller when the zone is undergoing rapid changes in the state variables and larger as the zone approaches steady state. Subroutine ZONE estimates a new time step each time it is called. The estimation is based on the requirement that the change in any state variable for one Euler time step must not exceed a selected maximum value. For example, in the building emulator model program, this maximum has been chosen to be 0.1 degrees F for temperature state variables. Since the value of the function which describes the right hand side of the differential equation (described in section 5.3.1) is the rate of

change of the state variable, the smallest desirable zone model time step may be determined by dividing the maximum allowable change in the state variable per time step by the current value of the right hand side function. Of course, calculations must be performed to ensure that the zone model time step is not larger than the emulator time step or smaller than a certain minimum. In the building emulator program zone model subroutine the time step can be no smaller than 0.001 seconds.

After the zone model time step is determined, it is adjusted so that it evenly divides the emulator time step. The quotient of this division is the number of times that the ZONE subroutine must call the EULER and SURFAC subroutines for each emulator model time step.

5.3.3 Differential Equation Function

The FORTRAN function ZONEFN is used by the zone model subroutine to calculate the current value of the right hand sides of the differential equations describing the state variables. Since there are three differential equations for the zone model, there are three possible right hand sides of the differential equations. The right hand side to be calculated is determined by an index which is 1 for the zone air energy balance, 2 for the zone moisture balance, and 3 for the wall energy balance. The expanded right hand sides of the equations corresponding to equations (5-1 through 5-3) are presented below. The variables which are designated by an upper case F with a subscript are implemented in the ZONEFN subroutine in a special way. Rather than being variables or constants, they are FORTRAN functions which return the value of a coefficient or energy term. In the definition of terms following each equation below, the FORTRAN function name used in the emulator model program will be listed. For the zone air energy balance:

$$\text{ZONEFN} = \frac{F_{sa} + F_I(T_o - T_i) + F_{cw}(T_{iw} - T_i) + F_{cg}(T_{ig} - T_i) + F_{ig}}{F_{cz}} \quad (5-8)$$

- where:
- F_{sa} : energy entering or removed from the zone by the air handling unit (function SUPPLY).
 - F_I : coefficient for infiltration of outdoor air into zone (function INFILT).
 - F_{cw} : coefficient for convection between the exterior wall surface and the zone air (function CONVWA).
 - F_{cg} : coefficient for convection between the surface of a window in the exterior wall and the zone air (function CONVGL).
 - F_{ig} : energy entering the zone due to internal gains from lights, equipment, and people (function INTERN).
 - F_{cz} : thermal mass of the zone air and furnishings (function MASSTH).

For the zone moisture balance:

$$\text{ZONEFN} = \frac{F_{sw} + F_{Im}(W_o - W_i) + F_{mg}}{F_{mz}} \quad (5-9)$$

- where: F_{sw} : moisture added to or removed from the zone by the air handling unit (function SUPPLW).
 F_{Im} : coefficient for infiltration of moisture (or exfiltration) from outdoor air into zone (function INFILW).
 F_{mg} : moisture entering the zone due to internal moisture gain from equipment and people (function MOIST).
 F_{mz} : effective mass of zone air and furnishings which can absorb moisture (function MASSA).

for the prediction of the rate of change of moisture in the zone, an additional check is made to insure that the zone air does not become oversaturated. This is accomplished by first using the function HUMRAT to determine the humidity ratio at the zone air temperature, T_i , and 100 percent relative humidity. If the current indoor humidity ratio, W_i , exceeds this value and ZONEFN is a positive value, then ZONEFN is set to zero, indicating that no further increase in zone moisture is possible. If W_i is zero and ZONEFN is negative, ZONEFN is set to zero indicating that zone moisture cannot be decreased since the air in the zone is completely dry.

The right hand side function for the exterior wall mass heat balance is:

$$\text{ZONEFN} = \frac{F_{co}(T_{sol} - T_w) + F_{ci}(T_{iw} - T_w)}{F_{cw}} \quad (5-10)$$

- Where: F_{co} : coefficient for conduction and convection heat transfer between the mass center of the wall and the outdoor air (function CNDCTO).
 F_{ci} : coefficient for conduction heat transfer between the mass center of the wall and the inner surface of the wall (function CNDCTI).
 T_{sol} : the sol-air temperature (defined in section 3.3.4)
 F_{cw} : thermal mass of the wall structure (function MASSW).

5.3.4 Surface Heat Balance Subroutine

The subroutine SURFAC is called by ZONE to solve the algebraic equations (5-4) and (5-5) for the inner surface temperatures of the exterior wall and glass window. The two equations after rearrangement to solve for the surface temperatures are shown below. The variables which are designated by an upper case F with a subscript are implemented in the SURFAC subroutine in a special

way. Rather than being variables or constants, they are FORTRAN functions which return the value of a coefficient or energy term. In the definition of terms following each equation below, the FORTRAN function name used in the emulator model program will be listed. For the exterior wall inner surface balance:

$$T_{iw} = \frac{F_{Aw} T_w + F_{Bw} T_i + F_{Rw} F_{Tr} + F_{ts}}{F_{Aw} + F_{Bw} + F_{Rw}} \quad (5-11)$$

Where: F_{Aw} : coefficient for conduction heat transfer between inner surface of wall and mass center of wall (function CNDCTI).
 F_{Bw} : coefficient for convection heat transfer between inner surface of wall and zone air (function CONVWA).
 F_{Rw} : coefficient for radiation heat transfer between inner surface of wall and zone radiant temperature (function WRADIA).
 F_{Tr} : mean radiant temperature of zone (function TRN).
 F_{ts} : energy which passes through windows in the zone and is absorbed at the wall surface (function WSOLAR).

For the window glass inner surface heat balance:

$$T_{ig} = \frac{F_{Ag} T_o + F_{Bg} T_i + F_{Rg} F_{Tr}}{F_{Ag} + F_{Bg} + F_{Rg}} \quad (5-12)$$

Where: F_{Ag} : coefficient for conduction and convection heat transfer between the inner surface of the glass and the outdoor air (function GCNDCT).
 F_{Bg} : coefficient for convection heat transfer between the inner surface of the glass and the zone air (function GCNVCT).
 F_{Rg} : coefficient for radiation heat transfer between the inner surface of the glass and the zone radiant temperature (function GRADIA).
 F_{Tr} : mean radiant temperature of zone (function TRN).

5.3.5 Fortran Coefficient Functions

The names of the special FORTRAN functions used to give coefficients or terms in the equations (5-8) through (5-12) were listed in the definitions following the equations. All of these functions have the same general form. The functions can be designated as either time-dependent or time-independent. Values of coefficients such as conduction coefficients may be considered time-independent and will need to be calculated only once. In this case, the function automatically determines if the function has already been evaluated and if so, returns the previously determined value.

Other coefficients such as a coefficient for infiltration heat transfer must usually be considered time-dependent. In this case the coefficient will be calculated each time the function is called unless the function is called twice for the same time step. In this case the function returns the previously determined value.

The general form of the FORTRAN function for a coefficient function is shown in table 5-1. The parameter TIMEDP is set to true for a time-dependent function and false for a time-independent function. The specific equations to evaluate the functions used in the emulator model program are listed in section 5.4.

5.4 Equations for Heat and Mass Transfer Coefficients and Terms

The following sections contain the equations used to evaluate the terms and coefficients to expand equations (5-8) through (5-12). The name of the FORTRAN coefficient function (see section 5.3.5) will be given in parenthesis following the title of the section. The equations listed would be placed in the area marked 'equations to evaluate function' in the generalized coefficient function listing in table 5-2.

Table 5-2. General form of the FORTRAN function for a coefficient function

```
FUNCTION [name](TIME)

LOGICAL TIMEDP, NOINIT
PARAMETER(TIMEDP = .TRUE.)
INTEGER OLD, TIME
SAVE NOINIT, [name], FUNC
DATA NOINIT/.TRUE./, OLD/-1/

IF(NOINIT.OR.(TIME.NE.OLD.AND.TIMEDP)) THEN
    FUNC = [ equations to evaluate function ]
    OLD = TIME
    NOINIT = .FALSE.
ENDIF
[name] = FUNC
RETURN
END
```

5.4.1 Sensible Heat Gain/Loss from Supply Air (SUPPLY)

The energy added to or removed from the zone by the supply air is determined from the difference in enthalpies of return and supply air. In addition, the

energy which is added to or removed from the room by the local zone heating or cooling equipment is included. The equation used is:

$$F_{sa} = h_{sa1} m_{sa1} + h_{sa2} m_{sa2} - h_{za} m_{ra} + Q_{lh} - Q_{lc} \quad (5-13)$$

Where: h_{sa1} : enthalpy of supply air
 m_{sa1} : mass flow rate of supply air
 h_{sa2} : enthalpy of hot deck supply air (optional)
 m_{sa2} : mass flow rate of hot deck supply air (optional)
 h_{za} : enthalpy of zone air
 m_{ra} : mass flow rate of return air
 Q_{lh} : energy added to zone by local zone heating equipment
 Q_{lc} : energy removed from zone by local zone cooling equipment

This function is considered to be time-dependent.

5.4.2 Coefficient for Sensible Heat Gain/Loss due to Infiltration (INFILT)

Sensible gain/loss due to infiltration is assumed to equal a coefficient multiplied by the difference in temperature between zone and outdoor air. The coefficient is determined from:

$$F_I = C_{p,inf1} \rho_{inf1} V_i I_{air} \quad (5-14)$$

where $C_{p,inf1}$: specific heat of infiltrated air
 ρ_{inf1} : density of infiltrated air
 I_{air} : air exchange rate (air changes/second)
 V_i : volume of zone space

The air exchange rate is considered to be a function of windspeed and indoor-outdoor temperature difference [Kusuda and Saitoh (1980), Achenbach and Coblenz (1963)]. The equation for air exchange rate per hour is:

$$I_{air} = \frac{I_{s,air}}{0.695} [0.15 + 0.013 V_w + 0.005 |T_o - T_i|] \quad (5-15)$$

where V_w : wind speed (mph)
 $I_{s,air}$: standard air exchange rate (model parameter, see table 2-8)
 = 1.5 for leaky building
 1.0 for standard building
 0.5 for modern type building

This function is considered to be time dependent.

5.4.3 Coefficient for Convective Heat Flow Rate from Walls (CONVWA)

The convective heat flow rate from the inner surface of the zone walls, including floor and ceiling, is assumed to be equal to a coefficient

multiplied by the difference between the wall surface temperature and the zone air temperature. The coefficient is given by:

$$F_{cw} = h_{iw,c} A_w \quad (5-16)$$

where $h_{iw,c}$: convective heat transfer coefficient at the interior surface
(model parameter, see table 2-8)
 A_w : surface area of the wall (model parameter, see table 2-8).

This coefficient is assumed to be time-independent.

5.4.4 Convective Heat Flow Rate from Glass Windows (CONVGL)

The convective heat flow rate from the inner glass surface of zone windows is assumed to be equal to a coefficient multiplied by the difference between the glass surface temperature and the zone air temperature. The coefficient is given by:

$$F_{cg} = h_{ig,c} A_g \quad (5-17)$$

where $h_{ig,c}$: convective heat transfer coefficient of inner glass
surface to zone air
 A_g : the glass window area

This coefficient is assumed to be time-independent. The abridged zone model effectively assumes that there are no windows in the zone, and therefore this coefficient is not used.

5.4.5 Sensible Internal Gains (INTERN) and Latent Internal Gains (MOIST)

5.4.5.1 Internal Gain Schedule Table

The FORTRAN functions used to determine internal gains, INTERN and MOIST, determine their values from a table which was created during the emulator model initialization. The table contains internal gain as a function of time-of-day. Interpolation is not used with the table and therefore the internal gain is not a continuous function of time. Internal gains remain at a constant level until changed by the next entry in the table at a particular time. The internal gain schedule table is stored in five FORTRAN arrays. The arrays contain time-of-day, sensible heat gain from equipment at the corresponding time-of-day, sensible heat gain from people, moisture gain from equipment and moisture gain from people. A function SCHED is used to determine a sensible or moisture gain from the internal gain tables. If moisture gain is requested (by MOIST), SCHED returns the sum of the moisture gain from people and equipment.

If a sensible gain is requested (by INTERN), SCHED returns the sum of the sensible gains from people and equipment.

5.4.5.2 Determining Values for the Internal Gain Schedule Table

The values in the internal gain schedule table are created during the initialization process of the emulator model. An example input parameter file was given in table 2-7. The following equations show how a row in the schedule table is created from the information in a line of table 2-7.

5.4.5.2.1 Convective Heat Gain from People

The internal gain from a person is assumed to depend only on the activity in which the person is engaged and not on the surrounding temperature. Table 5-3 is used as the basis for the sensible and moisture gains from people at various activity levels. The number of occupants at each activity level at a given time is part of the information entered in the building use parameters in table 2-7. The sensible heat generated by the occupants of a zone at a particular time is then assumed to be described by the following equation [Building Environment, NBS (1977), p. A-21]:

$$Q_{\text{people,c}} = (1 - r_p) \sum_{k=1}^{N_{\text{act}}} n_{p,k} W_{p,s,k} \quad (5-18)$$

where r_p : ratio of radiation energy to total sensible heat of people (=0.4)
 $n_{p,k}$: number of people who are doing the same (k-th) activity
 $W_{p,s,k}$: sensible heat generation by a person doing the k-th activity.
 N_{act} : number of activities

Table 5-3. Occupant sensible and moisture gains

[Table 18 entitled 'Rates of Heat Gain from Occupants of Conditioned Spaces' ASHRAE fundamentals (1981), p.26.25 and Threlkeld, page 363]

	sensible gain (Btu / hr.)	moisture gain (lb. water / hr.)
1. seated, very light work	230.	0.20
2. seated, light work (typing)	255.	0.27
3. standing, light work	315.	0.34
4. light bench work	345.	0.46
5. walking, 3 mph, light machine work	345.	0.74

5.4.5.2.2 Moisture Gain from People

$$M_{\text{people,lat}} = \sum_{k=1}^{N_{\text{act}}} n_{p,k} W_{p,\text{lat},k} \quad (5-19)$$

where N_{act} : number of activity
 $n_{p,k}$: number of people who are doing the same (k-th) activity
 $W_{p,\text{lat},k}$: moisture generation by a person doing the k-th activity

5.4.5.2.3 Convective Heat Gain from Equipment and Lighting

The total sensible internal gain from equipment and lighting is determined from the gains due to lighting added to the gain due to equipment. Parameters for the lighting power, lighting type, and equipment sensible gain at various times of the day are entered in the building use parameter file (see table 2-7, for example). For building equipment, the sensible internal gain is described by the following equation [ASHRAE (1981), pp. 26.26 - 26.29]:

$$Q_{\text{equip,c}} = (1 - r_e) W_{e,s} \quad (5-20)$$

where r_e : ratio of radiation energy to total equipment energy
 (=0.2 to 0.8)
 $W_{e,s}$: total sensible energy produced by equipment

The sensible internal gain from lighting is assumed to be described by the following equation:

$$Q_{\text{light,c}} = (1 - r_l) W_l \quad (5-21)$$

where W_l : total electric input
 r_l : ratio of radiation energy to total light energy
 = 0.8 for incandescent lights [Building Environment, NBS (1977), p. A-25]
 = 0.5 for fluorescent lights

5.4.5.2.4 Moisture Gain from Equipment

Moisture gain from equipment is specified in the parameters for the building use as shown in table 2-7. The equation for equipment moisture gain is: [ASHRAE (1981), p. 26.26 - 26.28]

$$M_{\text{equip,lat}} = W_{e,\text{lat}} \quad (5-22)$$

where $W_{e,\text{lat}}$ is the total moisture produced by equipment.

5.4.6 Thermal Mass of Zone Air and Furnishings (MASSTH)

The thermal mass which stores energy and dampens fluctuations in zone air temperature is contained in the air in the zone and the zone furnishings, such as desks, cabinets, and partitions. The shell parameter file, an example of which is given in table 2-8, contains the volume of the zone and the thermal capacity of the furnishings. The capacity of the zone air is given by:

$$C_{air} = C_{p,i} \rho_i V_i \quad (5-23)$$

where $C_{p,i}$: specific heat of moist zone air
 ρ_i : density of zone air
 V_i : volume of zone air (volume of interior space) (parameter, table 2-8)

The total thermal mass of the zone is then:

$$F_{cz} = C_{fur} + C_{air} \quad (5-24)$$

where C_{fur} : thermal mass of furnishings (parameter, table 2-8)

5.4.7 Moisture Gain/Loss due to Infiltration (INFILW)

Moisture gain/loss due to infiltration is assumed to equal a coefficient multiplied by the difference in humidity ratio between zone and outdoor air. The coefficient is determined from:

$$F_{Im} = \rho_{infl} V_i I_{air} \quad (5-25)$$

where ρ_{infl} : density of infiltrated air
 I_{air} : air exchange rate (see equation 5-15)
 V_i : volume of zone (parameter, table 2-8)

The air exchange rate is determined by the same equation as in section 5.4.2. The density of the infiltration air is determined by taking the average of the densities of the zone air and the outdoor air. This function is considered to be time-dependent.

5.4.8 Moisture Gain/Loss from Supply Air

The moisture added to or removed from the zone by the air handling unit is determined from the difference in the humidity ratios of return and supply air. The equation used is:

$$F_{sw} = W_{sa1} m_{sa1} + W_{sa2} m_{sa2} - W_{ra} m_{ra} \quad (5-26)$$

where: W_{sa1} : humidity ratio of supply air
 m_{sa1} : mass flow rate of supply air
 W_{sa2} : humidity ratio of hot deck supply air (optional)
 m_{sa2} : mass flow rate of hot deck supply air (optional)
 W_{ra} : humidity ratio of return air
 m_{ra} : mass flow rate of return air

This function is considered to be time-dependent.

5.4.9 Mass of Zone Air and Furnishings that can Absorb Moisture (MASSA)

The mass in the zone which stores moisture and dampens fluctuations in zone humidity ratio is contained in the air in the zone and the zone furnishings, such as curtains, paper, and wood furnishings. The shell parameter file, an example of which is given in table 2-8, contains the volume of the zone and a factor which is used to produce an effective air mass for moisture absorption by the combination of the air and the zone furnishings. This factor is difficult to measure since the moisture absorption properties of office materials are not only difficult to determine but depend on air flow and existing moisture content. This function is considered time-independent. The moisture absorbing capacity of the zone air and furnishings is described by the following equation:

$$F_{mz} = \rho_i V_i R_{mg} \quad (5-27)$$

where ρ_i : density of zone air
 V_i : volume of zone air (volume of interior space)
 R_{mg} : a factor used to produce the effective air mass for moisture absorption by the zone furnishings and air.

5.4.10 Overall Thermal Conductance between Wall and Outdoor Air (CNDCTO)

Heat transfer between the mass center of the wall and the outdoor air is a combination of conduction heat transfer within the wall and convection heat transfer from the wall outer surface to the outdoor air. The overall conductance is expressed by:

$$F_{co} = \frac{1}{\frac{1}{h_o} + \frac{L_{ow}}{k_{ow}}} A_w \quad (5-28)$$

where L_{ow} : thickness of wall between the mass center and the outer surface.
 k_{ow} : overall conductivity of the wall between the mass center and the outer surface.
 h_o : combined heat transfer coefficient (radiative + convective) for heat transfer between the outer wall surface and the outside air.
 A_w : surface area of the exterior wall.

The four parameters in this equation are obtained from the shell parameter file, exemplified by table 2-8. This is considered a time independent function.

5.4.11 Thermal Conductance between the Wall and Indoor Air (CNDCTI)

Heat transfer between the mass center of the wall and the inner surface of the wall is by conduction. The thermal conductance is expressed by:

$$F_{ci} = \frac{k_{iw}}{L_{iw}} A_w \quad (5-29)$$

where L_{iw} : thickness of wall between the mass center and the inner surface.
 k_{iw} : overall conductivity of the wall between the mass center and the inner surface.
 A_w : surface area of the exterior wall.

The three parameters in this equation are obtained from the shell parameter file, exemplified by table 2-8. This is considered a time independent function.

5.4.12 Thermal Capacitance of the Wall

The mass in the exterior wall which stores energy and dampens fluctuations in wall temperature is contained in the building materials of which the wall is constructed. The wall is divided into an inner and outer section by the mass center of the wall. The shell parameter file, an example of which is in table 2-8, contains the thickness, density, thermal conductivity and specific heat of the inner and outer wall sections. This function is considered time-independent. The thermal capacitance of the exterior wall is described by the following equation:

$$F_{cw} = (C_{p,ow} \rho_{ow} L_{ow} + C_{p,iw} \rho_{iw} L_{iw}) A_w \quad (5-30)$$

where $C_{p,ow}$, $C_{p,iw}$: average specific heat of outer and inner sections of wall.
 ρ_{ow} , ρ_{iw} : average density of outer and inner sections of wall.
 L_{ow} , L_{iw} : thickness of outer and inner sections of wall.
 A_w : area of the exterior wall.

5.4.13 Radiative Heat Transfer Coefficient (WRADIA)

Radiative heat transfer between the inner surface of the exterior wall and the other surfaces in the zone is assumed to be equal to a radiative heat transfer coefficient multiplied by the difference in the mean radiant temperature of the zone and the wall surface temperature. For simplicity, this mode of heat transfer was not implemented in the abridged building emulator model program. Determination of this mode of heat transfer involves the determination of view angle factors of surfaces which enclose the zone. [Walton (1980), Sowell and Walton (1980)]. Using the view angle, the radiant heat transfer coefficient could then be determined [Walton (1980), p. 69]. However, it is necessary to know the emissivities and the surface temperatures of all zone surfaces, as well as the specific geometry. The abridged model effectively assumes that the radiative coefficient is small or the difference between the surface and mean radiant temperatures is negligible.

5.4.14 Solar Energy Entering the Zone Absorbed by Walls (WSOLAR)

If the zone contains a glass window, solar energy may enter the zone through the window and be absorbed by the walls and furnishings. For simplicity, this mode of heat transfer was not implemented in the abridged building emulator model program. Since the only zone wall which has inner surface temperature as a state variable is the exterior wall, and this is the wall which is assumed to have a glass window, no solar energy from the window would be absorbed by the exterior wall. Solar energy could be absorbed by the furnishings, however. The abridged model effectively assumes that the zone walls are without windows. The equation for solar energy absorption is included in this section for completeness. If it is assumed that solar irradiation is not absorbed by the glass window:

$$F_{ts} = \frac{A_g r_g S_c I}{A_w} \quad (5-31)$$

where A_g : glass window area
 r_g : solar transmittance through glass [ASHRAE (1981), p. 27.38]
 I : total solar radiation on the glass window
 S_c : shading coefficient [ASHRAE (1981), p.27.38]

5.4.15 Thermal Conductance Through Glass to Outdoor Air (GCNDCT)

Heat transfer between the inner surface of the glass window and the outdoor air is a combination of conduction heat transfer within the glass and convection heat transfer from the glass outer surface to the outdoor air. This mode of heat transfer is not included in the abridged emulator model program since the abridged model assumes that there are no windows in the zone. The overall conductance equation is included for completeness and is expressed by:

$$F_{Ag} = \frac{1}{\frac{1}{h_o} + \frac{L_g}{k_g}} A_g \quad (5-32)$$

where L_g : thickness of glass.
 k_g : conductivity of the glass.
 h_o : combined heat transfer coefficient (radiative + convective) for heat transfer between the outer glass surface and the outside air.
 A_g : surface area of glass window.

5.4.16 Thermal Convection from Inner Glass Surface to Indoor Air (GCNVCT)

Heat transfer between the inner glass surface and the zone air is assumed to be equal to a convective heat transfer coefficient multiplied by the temperature difference between the inner glass surface and the zone air. The heat transfer coefficient must be supplied in the building shell parameter file. This mode of heat transfer is not included in the abridged version of the building emulator model program since the abridged model assumes that there are no windows in the zone.

5.4.17 Radiant Heat Transfer between Glass and Zone (GRADIA)

Radiative heat transfer between the inner surfaces of windows and the other surfaces in the zone is assumed to be equal to a radiative heat transfer coefficient multiplied by the difference in the mean radiant temperature of the zone and the glass surface temperature. Since there are assumed to be no windows in the zone, this mode of heat transfer was not implemented in the abridged building emulator model program. Determination of this heat transfer is equivalent to the process described in section 5.4.13.

5.4.18 Mean Radiation Temperature (MRT)

The mean radiant temperature is used to determine radiation heat transfer between the zone environment and wall or glass surfaces. The mean radiant temperature is a function of wall and glass surface temperatures, the radiant heat from people, the radiant heat from equipment, the properties of the

surfaces, and the geometry of surfaces [Walton (1983), p. 69]. Radiation heat transfer is not included in the abridged version of the emulator model program and the equation for mean radiant temperature is not included here.

5.5 Property Equations for Moist Air

The building emulator model program uses a number of functions to determine properties of moist air. The equations used are not presented in this report, but the names of the computer routines, the property determined, and a reference for the equations, is given in table 5-4.

5.6 Local Equipment Model

The local heating and cooling equipment is defined as the equipment located in or adjacent to the zone which is designed to directly heat or cool only that zone. The purpose of this equipment is to adjust the energy in the incoming air from the air handling unit to match the heating/cooling requirements of the zone. There are three different types of local equipment which may be present. These are: 1. energy consuming local equipment such as reheat coils, fan coils, heat pumps, perimeter radiation, or electric heaters; 2. air volume reduction controls to adjust the volume of supply air to match the zone requirements for VAV systems; and 3. air mixing controls to mix two air streams of different temperatures to match the zone requirements for multizone or dual-duct systems. Local equipment is usually under local control, which means that the controller for the equipment is located adjacent to the zone and the control sensor is located in the zone.

The subroutine LOCAL in the building emulator model program is used to simulate local equipment and determine energy usage required to maintain the zone air temperature as close to a local setpoint as possible. This subroutine is called by the ZONE subroutine for each zone model time step in the solution of the zone model equations.

Table 5-4 Moist air property routines

CPAIR : specific heat of moist air as a function of humidity ratio [ASHRAE fundamentals 1981, page 5.3]

RHOAIR : density of moist air as a function of humidity ratio and dry bulb temperature [ASHRAE fundamentals 1981, page 5.3]

ENTHAL : enthalpy of moist air as a function of dry bulb temperature, humidity ratio, and atmospheric pressure [ASHRAE fundamentals 1981, page 5.4]

TEMP : dry bulb temperature of moist air as a function of enthalpy, humidity ratio, and atmospheric pressure [ASHRAE fundamentals 1981, page 5.4]

DEWPT : dew point temperature of moist air as a function of humidity ratio, and atmospheric pressure [ASHRAE fundamentals 1981, page 5.4, equation 40b]

HUMRAT : Humidity ratio of moist air as a function of dry bulb temperature, relative humidity, and atmospheric pressure [ASHRAE fundamentals 1981, page 5.4]

RELHUM : Relative humidity of moist air as a function of dry bulb temperature, humidity ratio, and atmospheric pressure [ASHRAE fundamentals 1981, page 5.4]

PWSAT : Saturation pressure of water vapor as a function of absolute temperature [ASHRAE fundamentals 1981, page 5.2]

TDBSAT : dry bulb temperature of saturated air as a function of the enthalpy of saturated air and the atmospheric pressure [ASHRAE fundamentals 1981, page 6.5, table 2]

The zone model will require a value for the energy being added to or removed from the zone due to the HVAC system. This energy, designated as q_{ent} , which enters or leaves the zone, consists of two parts, the energy removed or added by the air from the air handling unit, q_{air} , and the energy removed or added by local energy consuming equipment, q_{local} . The q_{local} energy may not be present on certain systems. q_{air} can be calculated by:

$$q_{air} = m_{sa1} h_{sa1} + m_{sa2} h_{sa2} - h_i m_{ra} \quad (5-33)$$

where m_{sa} are supply air mass flow rates (1 indicates the cold deck and 2 the optional hot deck for a dual deck system), h_{sa} are the supply air enthalpies,

m_{ra} is the return air mass flow rate, and h_i is the interior space enthalpy of the zone.

The actions that the local equipment model simulates are those of the local controller which will control the local equipment to maintain a zone air temperature at a setpoint. If the actual zone air temperature deviates from the setpoint, the local controller will attempt to return the temperature to the setpoint by increasing or decreasing the local equipment output. A simple proportional plus integral control (PI-control) model will be used. For this model, the adjustment the local controller attempts to make to q_{ent} is determined from:

$$dq_{ent} = K_p (e - e_0) + K_I (e \, dt) \quad (5-34)$$

where e , error = $T_{setpoint} - T_i$, e_0 is the previous value of the error, K_I is an integral gain factor, K_p is a proportional gain factor, and dt is the zone model time step. Once dq_{ent} is determined, the type of HVAC system determines how q_{air} and q_{local} are to be determined.

If the system is dual deck, it is assumed that there is no local conditioning equipment beyond the mixing box for the two deck air streams. Then:

$$dq_{air} = dq_{ent} \quad (5-35)$$

$$q_{air}(t+dt) = q_{air}(t) + dq_{air} \quad (5-36)$$

The total mass flow rate of air from the air handling unit is assumed to be constant at m_{sa} . The supply air enthalpy after mixing must then be:

$$h_{sa} = \frac{q_{air}}{m_{sa}} + h_i \quad (5-37)$$

Since the flow rates of the two decks must sum to equal the total mass flow rate, and the two deck air streams are assumed to mix ideally and have a known supply air temperature, humidity ratio, and enthalpy, the unknown mass flow rates for the decks can be solved for:

$$m_{sa2} = \frac{m_{sa} (h_{sa} - h_{sa1})}{h_{sa2} - h_{sa1}} \quad (5-38)$$

$$m_{sa1} = m_{sa} - m_{sa2} \quad (5-39)$$

also: $W_{sa} = m_{sa1} W_{sa1} + m_{sa2} W_{sa2} \quad (5-40)$

If the system is a single deck VAV, it is assumed that there is no local

conditioning equipment beyond the VAV box for modulating the supply air flow rate into the zone. This may not be true for some systems, for example VAV reheat systems. Then:

$$dq_{air} = dq_{ent} \quad (5-41)$$

$$q_{air}(t+dt) = q_{air}(t) + dq_{air} \quad (5-42)$$

The enthalpy of the air from the air handling unit is assumed to be constant at h_{sa} . The mass flow rate, m_{sa} , can be varied. The supply air flow rate for the new value of energy addition to the zone must then be:

$$m_{sa} = \frac{q_{air}}{h_{sa} - h_i} \quad (5-43)$$

The local zone VAV box will have a maximum and minimum air flow rating. If these limits are exceeded, the air flow rate will be constrained to the limits. If this is the case, the energy entering the room will have to be recalculated based on the limited mass flow rate.

If the system is single deck constant volume, it is likely to have local conditioning equipment. In this case:

$$dq_{local} = dq_{ent} \quad (5-44)$$

$$q_{local}(t+dt) = q_{local}(t) + dq_{local} \quad (5-45)$$

If q_{local} exceeds the local capacity, it will remain at the local capacity. If q_{local} becomes negative, this represents a need for local cooling. If the local equipment has no cooling capacity, such as a reheat coil, then q_{local} will be set to zero. If cooling capacity is present, then q_{local} can take on negative values up to the cooling capacity of the local equipment.

6. OCCUPANT COMFORT MODEL

6.1 Assumptions and Purpose of Comfort Model

The purpose of the occupant comfort model is to determine in a basic way if the conditions in the zone are in accordance with ASHRAE standard 55-74 for thermal comfort. This standard basically assumes that comfort is acceptable if the zone dry bulb temperature and humidity ratio are within certain limits.

The test for comfort is only performed if the building zone is determined to be in an occupied state. The zone is assumed to be occupied if the internal gain schedule table, contained in the model function SCHED, predicts that the sensible gain from people is non-zero for the current time-of-day. (see section 5.4.5).

6.2 Comfort Envelope Equations

The equations to determine if the current state of zone air is within the ASHRAE comfort limits are based on the simple two-dimensional geometry of the psychometric chart. The upper and lower comfort limits for humidity ratio are constant and this allows a simple comparison of zone humidity ratio to the limits. If the humidity ratio in the zone is greater than 0.012 or less than 0.0044 pounds of water per pound of dry air then the zone is automatically outside of comfort limits.

The maximum and minimum comfort limits for dry bulb temperature are a linear function of humidity ratio between the upper and lower humidity ratio limits as expressed by:

$$T_{\text{lower limit}} = 72.58 - 131.58 W_{za} \quad \text{and} \quad (6-1)$$

$$T_{\text{upper limit}} = 81.73 - 394.74 W_{za} \quad (6-2)$$

where W_{za} is the zone humidity ratio in pounds of water per pound of dry air, and T is the dry bulb temperature in degrees Fahrenheit. If the zone dry bulb temperature is above or below these limits, then the zone is outside of comfort limits.

The total time that the zone is outside of comfort limits during occupied hours is determined by the comfort model subroutine. The maximum and minimum values of zone dry bulb temperature and relative humidity are also recorded.

7. COMPILATION OF ENERGY USE

The predicted energy used by the emulated building during a test run of the emulator is determined by the FORTRAN subroutine CMPILE. CMPILE uses the values of the temperature, enthalpy, and air flow rate variables in the state vector to calculate the energy used between the current and previous time steps in several categories, and essentially integrates the energy use by keeping cumulative totals which are placed in the state vector.

7.1 Categories of Energy Use

Energy used by a building is assumed to be categorized as follows:

1. energy used by the fans in the air handling unit (the power used by the fans will be constant unless a variable air volume system is used, in which case fan power will vary as a function of air flow).
2. energy used to heat or cool the air passing through the air handling unit (this would not include cooling by outside air).
3. energy used by local space heating or cooling equipment (reheat coils, heat pumps, fan coils, perimeter radiation, air conditioners).

It is assumed that the calculated energy values are not in terms of any fuel and do not include any plant efficiencies. The energies are either in energy added to the air and extracted from steam or hot water for heating, or in energy required to be mechanically extracted from the air by chilled water or refrigerant for cooling.

There are 4 primary sources of reduction in energy use which EMCS algorithms may cause. Values for the energy reductions may be calculated from the above types of energy use in a building. These are:

1. savings from not running electrical loads (fans, pumps).
2. savings from reducing the daily heating or cooling requirements of a building by reduction of operating hours or changing of local space temperature setpoints.
3. savings by using the outside air for heating or cooling and avoiding mechanical heating or cooling.
4. savings from reduction of reheating or recooling of air passing through the air handler and building space for purposes of humidity control and compensation for load variation between zones. These causes of energy consumption are sometimes termed system effects, since the amount of energy for this purpose will vary between different HVAC systems, such as between reheat and multizone.

7.2 Equations for Energy Use

The first category of energy consumption is air handling unit fan energy consumption. If the system is constant volume, the steady-state power consumption of the fan can be used to calculate energy. If the system is variable air volume, it is assumed that the fan power consumption will vary with the volume flow rate through the unit. If the rated power consumption of a fan is known at a rated flow rate, then the power consumption at another flow rate can be determined based on fan law number 3 from the ASHRAE equipment handbook [1983, page 3.5, table 2]:

$$P_{fan} = P_{rated} \left(V_a / V_{rated} \right)^3 \quad (7-1)$$

where: P_{rated} : power consumption of fan at rated air flow rate.
 V_{rated} : rated fan volumetric air flow rate.
 V_a : current volumetric flow rate.

The fan energy between the current and last time step is determined by multiplying P_{fan} by the emulator model time step. The cumulative energy for the fan is determined by adding the current time step energy to the sum of values of energy for previous time steps.

The thermal energy used for heating and cooling the building should be divided into the energy added to the building air for heating purposes and the energy removed from the building air for cooling purposes. Therefore the energy values will always have two parts, heating and cooling. The summation or subtraction of two energies will then consist of the separate summation or subtraction of the heating and cooling parts (this is analogous to two-dimensional vectors with x and y components). For the following equations, an energy rate is calculated. This rate is multiplied by the emulator model time step to determine the energy used since the last time step. The cumulative energy is obtained by adding this to the sum of energy values of previous time steps.

The requirements energy is the energy required to bring the air leaving the building zones through the return ducts and the ventilation air to the level of the supply air enthalpy (including both air streams if dual deck).

The requirements energy rate (power) is:

$$q_{req} = m_{sal} (h_{sal} - h_{mc}) \quad (7-2)$$

where the mc subscript indicates the mixed air temperature in the air handling unit with only minimum outside air or ventilation air (see section 4.2). If $q_{req} < 0$ this is cooling energy, if $q_{req} > 0$ this is heating energy. If the system is dual deck there is an additional component of the requirements energy rate for the hot deck.

$$q_{req2} = m_{sa2} (h_{sa2} - h_{mc}) \quad (7-3)$$

If $q_{req2} < 0$ this is cooling energy, if $q_{req2} > 0$ this is heating energy. The hot deck component is added to the cold deck component of requirements energy, keeping heating and cooling energies separated.

The actual energy is the energy required to bring the air passing through the air handler to the level of the supply air enthalpy from the level of the mixed air enthalpy. This is usually equal to the energy supplied or removed by the heating and cooling coils. This actual energy will be less than the requirements energy since part of the requirements energy can be met by an economizer algorithm using outdoor air for cooling.

The actual energy rate (q_{act1} for single or cold deck and q_{act2} for hot deck if a dual deck system) is then:

$$q_{act1} = m_{sal} (h_{sal} - h_{ma}) \quad (7-4)$$

$$q_{act2} = m_{sa2} (h_{sa2} - h_{ma}) \quad (7-5)$$

q_{act1} and q_{act2} are summed, keeping heating and cooling energies separated.

The requirements and actual energy can be used to determine the economizer energy. This is not usually considered an energy use, since the outdoor air costs nothing to condition. However, for purposes of calculating energy savings for economizer algorithms (savings category 3 above) it is useful to consider this as energy consumption. The economizer energy rate is:

$$q_{econ} = q_{req} - q_{act} \quad (7-6)$$

The actual energy can be divided into two parts which are the other two categories of building energy use. The energy that would be required to bring the mixed air to the actual condition of the air entering the building space (after mixing for a dual deck system) can be termed the load energy. This will equal the actual energy for all systems except dual deck. This represents the energy requirements with system effects minimized (used for savings in category 2, above). The load energy is:

$$q_{load} = (m_{sal} h_{sal} + m_{sa2} h_{sa2}) - (m_{sal} + m_{sa2}) h_{ma} \quad (7-7)$$

The final category of energy use is the energy used for reheating and recooling of air (used for savings in category 4, above). This is defined as:

$$q_{rhc} = q_{act} - q_{load} + q_{local} \quad (7-8)$$

This sum is of the heating and cooling components of the energies.

When a test run using the emulator is complete, the emulator model program will generate a summary report file containing energy usage for the emulated building over the test period. This task is performed by subroutine REPORT. The report will contain energy usage in the categories discussed above for heating and cooling energy components. An example energy report is given in table 7-1. The report file will also contain results on the amount of time the zone was not within comfort limits and the maximum and minimum space temperatures and relative humidities reached during occupied hours.

Table 7-1. Heating and cooling energy from emulator model

	HEATING		COOLING
FAN ENERGY			
ECONOMIZER ENERGY			
LOAD ENERGY			
REHEAT/RECOOL ENERGY			

8. SUMMARY

The use of a building emulator is one of the approaches to testing of EMCS systems without making actual connections to HVAC equipment or a building. The NCEL/NBS emulator will be used for either factory or field acceptance tests of EMCS algorithms. The building simulation program in the emulator is the main part of this report. This program simulates the responses of a building to EMCS control actions.

The building emulator program was written in FORTRAN 77 and the source code is appended in the APPENDIX. State variables in the emulator model are analog sensor, energy, command, and comfort variables. The building emulator model program contains the weather (WEATHR), air handling unit (AHU), zone (ZONE), and comfort (COMFRT) model. The use of energy during a simulation is accumulated by the energy compilation subroutine (COMPILE). This energy use information is useful to evaluate the performance of an EMCS algorithm.

Prior to simulation, data files of initial values of state variables, building shell parameters, HVAC system parameters, and weather data parameters must be called. The air handling unit model is a steady-state model, while the zone model is a dynamic model. In the zone model, a system of first order ordinary differential equations is solved by the explicit forward difference scheme. The time step for the difference scheme is adjusted according to the changes of state variables. By using this adaptive time step, which is uniformly distributed in a given emulator simulation time period, savings in computation time can be realized when the state variables approach steady state.

The air handling unit model simulates a single deck (or single duct), constant volume system without humidity control as shown in Figure 4-2. Damper positions of the air handling unit are varied if economizer option is selected. A single zone with exterior walls without window is modeled under the assumptions described previously. For this abridged system, a local equipment model is implemented. Heating/cooling energy delivered from the local equipment to zone air is adjusted under the control law of the proportional plus integral controller (PI-controller). Capabilities of simplified current models can be upgraded by providing necessary information in the spaces already reserved for future extension.

9. REFERENCES

ASHRAE, 1981 Fundamentals Handbook, ASHRAE, 1981.

Berglund, L., 'Control and Simulation of Thermal Comfort,' Workshop on HVAC Control, Modeling, and Simulation, Atlanta, GA, Feb. 2-3, 1984.

Borresen, B.A., 'Thermal Room Models for Control Analysis,' ASHRAE Trans., Vol. 87, Part 2, 1981.

Brokaw, R.S., 'Calculation of Flue Losses for High-Efficiency Furnaces and Appliances,' ASHRAE Journal, Jan. 1979, pp. 49-51.

Building Environment Division of NBS, 'Technical Guidelines for Energy Conservation,' AFCEC-TR-77-12 and NBSIR 77-1238, Nat'l Bureau of Standards, June 1977.

Bullock, C.E., 'Dynamic Simulation Models for Commercial Air Conditioning and Heat Pump Systems,' Workshop on HVAC Control, Modeling, and Simulation, Atlanta, GA, Feb. 2-3, 1984.

Chi, J., 'Building Environmental Control System Analysis Computer Program - Volume 1 Component Mathematical Models,' Final Report submitted to NBS, September 1982.

Fanger, P.O., Thermal Comfort, McGraw-Hill, 1970.

Ferziger, J.H., Numerical Methods for Engineering Application, John Wiley, 1981.

Kao, J.Y., 'Strategies for Energy Conservation for a Large Office Building,' NBSIR 83-2746, Nat'l Bureau of Standards, July 1983.

Kusuda, T., 'NBSLD, The Computer Program for Heating and Cooling Loads in Buildings,' NBS BSS 69, Nat'l Bureau of Standards, 1978.

May, W.B., 'Time of Day Control and Duty Cycling Algorithms for Building Management and Control Systems,' NBSIR 83-2713, Nat'l Bureau of Standards, June 1983.

May, W.B., 'Control Algorithms for Building Management and Control Systems-- Hot Deck/Cold Deck/Supply Air Reset, Day/Night Setback, Ventilation Purging, and Hot and Chilled Water Reset,' NBSIR 83-2713, Nat'l Bureau of Standards, March 1984.

Modest, M.F. and Macken, N., 'Seasonal Performance Comparisons for an Apartment Building in New York State Climate Regions,' ASHRAE Trans., 1977, pp. 881-892.

- Park, C. 'Single-zone Computer Model for Residential Furnace Location Analysis,' ASHRAE Trans., Vol. 87, Part 2, 1981.
- Park, C., Kelly, G.E. and Kao, J.Y. 'Economizer Algorithms for Energy Management and Control Systems,' NBSIR 84-2832, Nat'l Bureau of Standards, Feb. 1984.
- Park, C., 'An Optimum Start/Stop Control Algorithm for Heating and Cooling Systems in Buildings,' NBSIR 83-2720, Nat'l Bureau of Standards, May 1983.
- Park, C., 'Demand Limiting Algorithms for Energy Management and Control Systems,' NBSIR 84-2826, Nat'l Bureau of Standards, February 1984.
- Sowell, E.F. and Walton, G.N., 'Efficient Computation of Zone Loads,' ASHRAE Trans., Vol. 86, Part 1, 1980.
- Threlkeld, J.L., Thermal Environmental Engineering, 2nd ed., Prentice-Hall, 1970.
- Walton, G.N., 'A New Algorithm for Radiant Interchange in Room Loads Calculations,' ASHRAE Trans., Vol. 86, Part 2, 1980.
- Walton, G.N., 'Thermal Analysis Research Program Reference Manual,' NBSIR 83-2655, Nat'l Bureau of Standards, March 1983.
- Wise, B.B., 'Energy Monitoring and Control System Software Testing Device,' Ninth Energy Management & Controls Society Conference Proceedings, Nov. 1984.

APPENDIX

COMPUTER PROGRAM LISTINGS

```

1 C Version 1.7 - FEBRUARY 05, 1985 - W.B. MAY ,NATIONAL BUREAU OF STANDARDS
2 C=====
3 PROGRAM ALGTST
4 C=====
5 C This is the main program used to develop and test simulation routines
6 C for a simple building model to be used in an EMCS algorithm tester
7 C developed by the U.S. Navy Civil Engineering Laboratory. This test
8 C program takes the place of the EMCS tester software. Subroutine
9 C EMODEL is the emulator model. Subroutine UPDATE represents the I/O
10 C task which determines if any commands have arrived from the EMCS and
11 C updates the analog and digital signals to the EMCS based on the
12 C simulation calculations. It is assumed that these routines will
13 C eventually run in parallel, UPDATE at a higher frequency. Parallel
14 C execution is approximated by serial calls, but with UPDATE being
15 C called more often.
16 C
17 C Determine start time
18 CALL SYNCCL
19 C-----
20 C Update commands and outputs
21 C
22 1000 CALL UPDATE
23 C-----
24 C Call the emulator model
25 C
26 CALL EMODEL
27 C-----
28 C Update clock
29 C
30 CALL ELOCK
31 GO TO 1000
32 END
33 C=====
34 BLOCK DATA COMDAT
35 C=====
36 C This module contains a list of all common block data and globals which
37 C are referenced in the support routines for the EMODEL subroutine.
38 C Additional common blocks used only with EMODEL are described in the
39 C subroutine EMODEL or the subroutine PARAMA.
40 C common blocks are:
41 C ASENSR - contains all analog sensor readings in arrays. Sensor
42 C readings are copied from the emulator model outputs after
43 C each time step. Array index indicates which of 5 possible
44 C buildings the reading is from.
45 C COMMND - contains all command digital inputs from the EMCS in arrays.
46 C readings are directly available to the emulator model.
47 C CPA - contains all control point adjustments from the EMCS in arrays.
48 C DSENSR - contains all digital sensor readings in arrays.
49 C EVENT - GLOBAL which is a logical variable indicating an EMCS command.
50 C PAST - contains day, hour, minute, and second that model subroutine
51 C PFILES - contains the names of all parameter files, and the season.
52 C was last called, and the simulation major timestep.
53 C STATE1 - contains the state variables for the emulator model.
54 C TIME - current time is broken into DAY, HOUR, MINUTE, SECOND. In this
55 C test program, all times are zero at the start of the program.
56 C the basic time step of the program is one second. ENDDAY,ENDHR,
57 C ENDMIN, and ENDSEC are the scheduled time to end the simulation.

```

```

58 C UNITS - contains the logical units used by the program for various
59 C inputs and outputs.
60 C -----
61 C
62 C TOUTA - EMCS analog input, Temperature, Outside Air (F)
63 C RHOUTA - EMCS analog input, Relative Humidity, Outside Air (%)
64 C TZONEA - EMCS analog input, Temperature, Space (F)
65 C RHZONA - EMCS analog input, Relative Humidity, Space (%)
66 C TRET A - EMCS analog input, Temperature, Return Air (F)
67 C RHRETA - EMCS analog input, Relative Humidity, Return Air (%)
68 C ZHCDEM - EMCS analog input, Position, Reheat Valve (Btu/s)
69 C TSUPA - EMCS analog input, Temperature, Supply Air (F)
70 C TMIXA - EMCS analog input, Temperature, Mixed Air (F)
71 C
72 REAL TOUTA,RHOUTA,TZONEA, RHZONA, TRET A,
73 & RHRETA, ZHCDEM, TSUPA ,TMIXA
74 COMMON / ASENSR / TOUTA,RHOUTA,TZONEA(5),RHZONA(5),TRET A(5),
75 & RHRETA(5),ZHCDEM(5),TSUPA(5),TMIXA(5)
76 DATA TOUTA,RHOUTA,TZONEA,RHZONA,TRET A,RHRETA,ZHCDEM/27*0.0/
77 DATA TSUPA, TMIXA/10*0.0/
78 C
79 C ON - EMCS command, Supply Fan On/Off
80 C ECON - EMCS command, G.A. Damper Off/Auto
81 C VENT - EMCS command, Ventilation Damper Open/Close
82 C SETBAK - EMCS command, Zone Thermostat Setback On/Off
83 C
84 LOGICAL ON, ECON, VENT, SETBAK
85 COMMON / COMMND / ON(5),ECON(5),VENT(5),SETBAK(5)
86 DATA ON,ECON,VENT,SETBAK/20*.FALSE./
87 C
88 C SUPPLY - EMCS CPA, Supply Air Temperature (F)
89 C ZONE EMCS CPA, Zone Temperature (F)
90 C
91 REAL SUPPLY ,ZONE
92 COMMON / CPA / SUPPLY(5),ZONE(5)
93 DATA SUPPLY,ZONE/10*0.0/
94 C
95 C DPSSF - EMCS digital input, differential Pressure Switch, Supply Fan
96 C
97 LOGICAL DPSSF
98 COMMON / DSENSR / DPSSF(5)
99 DATA DPSSF/5*.FALSE./
100 C
101 C EVENT - Logical variable which is true when a command from the EMCS
102 C file has been received.
103 C
104 LOGICAL EVENT
105 GLOBAL EVENT
106 C
107 C OLDDAY - Day number when EMODEL subroutine was last called.
108 C OLDHR - Hour of day when EMODEL subroutine was last called.
109 C OLDMIN - Minute of hour when EMODEL subroutine was last called.
110 C OLDSEC - Second of minute when EMODEL subroutine was last called.
111 C STEP - Major time step for simulation. model routines for AHU ,
112 C weather, zone, comfort, and energy compilation are called
113 C at this interval. Not the same as the Euler timestep used
114 C within the zone model.
115 C

```

```

116         INTEGER          OLDDAY,OLDHR,OLDMIN,OLDSEC,STEP
117         COMMON / PAST    / OLDDAY,OLDHR,OLDMIN,OLDSEC,STEP
118         DATA OLDDAY,OLDHR,OLDMIN,OLDSEC,STEP/4*0,30/
119
120 C HVFILE - Character filename of HVAC equipment parameter file.
121 C USFILE - Character filename of Building usage parameter file.
122 C CLFILE - Character filename of climate description parameter file.
123 C SHFILE - Character filename of building shell parameter file.
124 C INFILE - Character filename of state vector initialization file.
125 C SEASON - Character name of weather season to use in test.
126
127         CHARACTER*15      HVFILE,USFILE,CLFILE,SHFILE,INFILE,SEASON
128         COMMON / PFILES  / HVFILE,USFILE,CLFILE,SHFILE,INFILE,SEASON
129
130 C A      - Real array containing state variables at the current time.
131 C L      - Logical array containing current logical state variables.
132
133         LOGICAL          L
134         REAL             A
135         COMMON / STATE1 / A(100);L(100)
136         DATA A/100*0.0/,L/100*.FALSE./
137
138 C DAY    - Current day number, first day of test is day 1.
139 C HOUR   - Current hour of day, from 0 - 23.
140 C MINUTE - Current minute within the hour, 0 - 59.
141 C SECOND - Current second within the minute, 0 - 59.
142 C ENDDAY - Day number on which the test is to stop.
143 C ENDHR  - Hour of the day on ENDDAY within which test is to stop.
144 C ENDMIN - Minute within ENDHR within which test is to stop.
145 C ENDSEC - Second within ENDMIN within which test is to stop.
146
147         INTEGER          DAY,HOUR,MINUTE,SECOND,ENDDAY,ENDHR,ENDMIN,ENDSEC
148         COMMON / TIME    / DAY,HOUR,MINUTE,SECOND,ENDDAY,ENDHR,ENDMIN,ENDSEC
149
150 C CLU    - Logical Unit for writes to console.
151 C FLU    - Logical Unit for read from EMCS command input file.
152 C ILU    - Logical Unit for read from console.
153 C PLU    - Logical Unit for read from parameter files.
154
155         INTEGER          CLU,FLU,ILU,PLU
156         COMMON / UNITS   / CLU,FLU,ILU,PLU
157         DATA CLU/1/,FLU/8/,ILU/1/,PLU/9/
158
159         END
160
161         SUBROUTINE SYNCCL
162
163 C This routine is used to determine the start time of the test. In a
164 C real-time system, this routine would set the real time clock to the
165 C proper time.
166
167 C Input variables: none
168 C Output variables: none
169 C Procedures called: none
170 C Common blocks: UNITS,TIME
171
172         INTEGER          CLU,FLU,ILU,PLU
173         COMMON / UNITS   / CLU,FLU,ILU,PLU

```

```

174 C
175     INTEGER          DAY, HOUR, MINUTE, SECOND, ENDDAY, ENDHR, ENDMIN, ENDSEC
176     COMMON / TIME / DAY, HOUR, MINUTE, SECOND, ENDDAY, ENDHR, ENDMIN, ENDSEC
177 C
178 C Local Variables:
179 C CFFILE - Character name of file containing configuration information.
180 C
181     CHARACTER*15 CFFILE
182     PARAMETER (CFFILE = 'BLDG.CNF')
183 C
184 C-----
185 C Read the configuration file to determine the start time.
186 C
187     OPEN(PLU, FILE=CFFILE, STATUS='OLD', ERR=9800)
188     READ(PLU, *) HOUR, MINUTE, SECOND
189     CLOSE(PLU)
190     DAY = 1
191     RETURN
192 C-----ERRORS-----
193     9800 WRITE(1,1)
194     1 FORMAT(1X, 'Cannot successfully open configuration file')
195     STOP
196     END
197 C=====
198     SUBROUTINE CONPAR
199 C=====
200 C This routine is used to determine all of the emulator routine control
201 C parameters including duration of test, parameter file
202 C names and weather season.
203 C
204 C Input variables: none
205 C Output variables: none
206 C Procedures called: none
207 C Common blocks: UNITS, TIME, PAST, PFILES
208 C
209     INTEGER          CLU, FLU, ILU, PLU
210     COMMON / UNITS / CLU, FLU, ILU, PLU
211 C
212     INTEGER          DAY, HOUR, MINUTE, SECOND, ENDDAY, ENDHR, ENDMIN, ENDSEC
213     COMMON / TIME / DAY, HOUR, MINUTE, SECOND, ENDDAY, ENDHR, ENDMIN, ENDSEC
214 C
215     INTEGER          OLDDAY, OLDHR, OLDMIN, OLDSEC, STEP
216     COMMON / PAST / OLDDAY, OLDHR, OLDMIN, OLDSEC, STEP
217 C
218     CHARACTER*15     HVFILE, USFILE, CLFILE, SHFILE, INFILE, SEASON
219     COMMON / PFILES / HVFILE, USFILE, CLFILE, SHFILE, INFILE, SEASON
220 C
221 C Internal Variables:
222 C
223 C CFFILE - Character name of file containing configuration information.
224 C
225     CHARACTER*15 CFFILE
226     PARAMETER (CFFILE = 'BLDG.CNF')
227 C
228 C-----
229 C Read the configuration file to determine the test duration, HVAC type
230 C file, building use file, weather data file, building type file,
231 C initialization file, season, and simulation time step.

```

```

232      C
233      OPEN(PLU,FILE=CFFILE,STATUS='OLD',ERR=9800)
234      READ(PLU,*)
235      READ(PLU,*) ENDDAY,ENDHR,ENDMIN,ENDSEC
236      READ(PLU,*) HVFILE
237      READ(PLU,*) USFILE
238      READ(PLU,*) CLFILE
239      READ(PLU,*) SHFILE
240      READ(PLU,*) INFILE
241      READ(PLU,*) SEASON
242      READ(PLU,*) STEP
243      CLOSE(PLU)
244      C-----
245      C Compute ending time
246      C
247      ENDDAY = ENDDAY + DAY
248      ENDSEC = ENDSEC + SECOND
249      IF(ENDSEC.GT.59)THEN
250          ENDSEC = 0
251          ENDMIN = ENDMIN + 1
252      ENDIF
253      ENDMIN = ENDMIN + MINUTE
254      IF(ENDMIN.GT.59)THEN
255          ENDMIN = 0
256          ENDHR = ENDHR + 1
257      ENDIF
258      ENDHR = ENDHR + HOUR
259      IF(ENDHR.GT.23)THEN
260          ENDHR = 0
261          ENDDAY = ENDDAY + 1
262      ENDIF
263      OLDDAY = DAY
264      OLDHR = HOUR
265      OLDMIN = MINUTE
266      OLDSEC = (-STEP)
267      RETURN
268      C-----ERRORS-----
269      9800 WRITE(1,1)
270      1 FORMAT(1X,'Cannot successfully open configuration file')
271      STOP
272      END
273      C=====
274      SUBROUTINE ECTRL(TIMEST,GO)
275      C=====
276      C This routine is used to control the execution timing of the emulator
277      C model routine. If GO is returned as TRUE, the emulator simulation
278      C should be executed. This occurs when 1. the time step duration has
279      C passed, and 2. a control event has been detected. TIMEST contains the
280      C time step that the model is to use in simulation.
281      C
282      C Input variables: none
283      C
284      C Output variables:
285      C TIMEST - The current value to be used for the major simulation time step.
286      C GO      - A logical variable which indicates that the simulation
287      C          subroutine should be executed if true.
288      LOGICAL GO
289      INTEGER TIMEST

```



```

290 C
291 C Procedures called: NUSTEP, TIMINC
292 C
293 C Common blocks: none
294 C
295 C Global Variables:
296 C EVENT - Logical variable which is true when a command from the EMCS
297 C file has been received.
298 C
299 C LOGICAL EVENT
300 C GLOBAL EVENT
301 C
302 C Local Variables:
303 C STEPDN - Logical variable, true if time for another simulation execution.
304 C TIMEUP - Logical variable, true if test ending time has been reached
305 C
306 C LOGICAL TIMEUP,STEPDN
307 C
308 C -----
309 C Update the time and determine if the execution timestep has passed.
310 C If so, STEPDN will be true. Also determine if the ending time has
311 C been reached. If so, TIMEUP will be true.
312 C
313 C CALL TIMINC(STEPDN,TIMEUP)
314 C IF(TIMEUP) THEN
315 C CALL REPORT
316 C STOP 'Emulation Complete'
317 C ENDIF
318 C IF(EVENT.OR.STEPDN)THEN
319 C GO = .TRUE.
320 C STEPDN = .FALSE.
321 C CALL NUSTEP(TIMEST)
322 C ELSE
323 C GO = .FALSE.
324 C ENDIF
325 C RETURN
326 C END
327 C=====
328 C SUBROUTINE NUSTEP(TIMEST)
329 C=====
330 C This routine returns the amount of time in seconds since the last time
331 C it was called.
332 C
333 C Input variables: none
334 C Output variables:
335 C TIMEST - Time in seconds since routine was last called.
336 C
337 C INTEGER TIMEST
338 C
339 C Procedures called: SBTIME
340 C Common blocks: TIME,PAST
341 C
342 C INTEGER DAY, HOUR, MINUTE, SECOND, ENDDAY, ENDHR, ENDMIN, ENDSEC
343 C COMMON / TIME / DAY, HOUR, MINUTE, SECOND, ENDDAY, ENDHR, ENDMIN, ENDSEC
344 C
345 C INTEGER OLDDAY, OLDHR, OLDMIN, OLDSEC, STEP
346 C COMMON / PAST / OLDDAY, OLDHR, OLDMIN, OLDSEC, STEP
347 C

```

```

348 C Local Variables: none
349 C
350 C----- determine time since last call -----
351 CALL SBTIME(DAY, HOUR, MINUTE, SECOND, OLDDAY, OLDHR, OLDMIN, OLDSEC,
352 & TIMEST)
353 C----- reset last time called variables -----
354 OLDDAY=DAY
355 OLDHR=HOUR
356 OLDMIN=MINUTE
357 OLDSEC=SECOND
358 RETURN
359 END
360 C=====
361 SUBROUTINE ECLOCK
362 C=====
363 C This routine increments the time. In a real-time system, the time would
364 C be obtained from a real-time clock..
365 C
366 C Input variables: none
367 C Output variables: none
368 C Procedures called: none
369 C Common blocks: TIME
370 C
371 INTEGER DAY, HOUR, MINUTE, SECOND, ENDDAY, ENDHR, ENDMIN, ENDSEC
372 COMMON / TIME / DAY, HOUR, MINUTE, SECOND, ENDDAY, ENDHR, ENDMIN, ENDSEC
373 C
374 C Local Variables: none
375 C
376 C----- Increment time -----
377 SECOND = SECOND + 1
378 IF(SECOND.GE.60)THEN
379 SECOND = 0
380 MINUTE = MINUTE + 1
381 IF(MINUTE.GE.60)THEN
382 MINUTE = 0
383 HOUR = HOUR + 1
384 IF(HOUR.GE.24)THEN
385 HOUR = 0
386 DAY = DAY + 1
387 ENDIF
388 ENDIF
389 ENDIF
390 RETURN
391 END
392 C=====
393 SUBROUTINE TIMINC(STEPDN, TIMEUP)
394 C=====
395 C This routine determines if the
396 C desired duration of the execution has passed (TIMEUP=true), and if the
397 C simulation timestep has passed since the last execution of the
398 C simulation subroutine (STEPDN=true).
399 C
400 C Input variables: none
401 C Output variables:
402 C STEPDN - Logical variable, true if time for another simulation execution.
403 C TIMEUP - Logical variable, true if test ending time has been reached
404 C
405 LOGICAL STEPDN, TIMEUP

```

```

406 C
407 C Procedures called: none
408 C Common blocks: TIME,PAST
409 C
410     INTEGER DAY, HOUR, MINUTE, SECOND, ENDDAY, ENDHR, ENDMIN, ENDSEC
411     INTEGER OLDDAY, OLDHR, OLDMIN, OLDSEC, STEP
412     COMMON / TIME / DAY, HOUR, MINUTE, SECOND, ENDDAY, ENDHR, ENDMIN, ENDSEC
413     COMMON / PAST / OLDDAY, OLDHR, OLDMIN, OLDSEC, STEP
414 C
415 C Local Variables:
416 C
417 C JTIME - Integer used to contain time differential in seconds.
418 C
419     INTEGER JTIME
420 C
421 C----- check for end of program -----
422     CALL SBTIME(ENDDAY, ENDHR, ENDMIN, ENDSEC, DAY, HOUR, MINUTE, SECOND,
423     & JTIME)
424     IF(JTIME.LT.0) THEN
425         TIMEUP = .TRUE.
426     ELSE
427         TIMEUP = .FALSE.
428     ENDIF
429 C----- check for end of time step -----
430     CALL SBTIME(DAY, HOUR, MINUTE, SECOND, OLDDAY, OLDHR, OLDMIN, OLDSEC,
431     & JTIME)
432     IF(JTIME.GE.STEP) THEN
433         STEPDN = .TRUE.
434     ELSE
435         STEPDN = .FALSE.
436     ENDIF
437     RETURN
438     END
439 C=====
440     SUBROUTINE SBTIME(DAY, HOUR, MIN, SEC, MDAY, MHOUR, MMIN, MSEC, DSEC)
441 C=====
442 C This module subtracts two sets of time integers (xxx - Mxxx) and
443 C puts the result in seconds into DSEC.
444 C
445 C Input variables:
446 C DAY - Day number of first time from which second time is subtracted.
447 C HOUR - Hour of first time.
448 C MIN - Minute of first time.
449 C SEC - Second of first time.
450 C MDAY - Day number of second time which is to be subtracted from first.
451 C MHOUR - Hour of second time.
452 C MMIN - Minute of second time.
453 C MSEC - Second of second time.
454
455 C Output variables:
456 C DSEC - differential in seconds between first and second times.
457 C
458     INTEGER DAY, HOUR, MIN, SEC, MDAY, MHOUR, MMIN, MSEC, DSEC
459 C
460 C Procedures called: none
461 C Common blocks: none
462 C
463 C Local Variables:

```

```

464 C
465 C CDAY - Working copy of day from first time.
466 C CHOUR - Working copy of hour from first time.
467 C CMIN - Working copy of minute from first time.
468 C CSEC - Working copy of second from first time.
469 C DDAY - time differential, day component.
470 C DHOUR - time differential, hour component.
471 C DMIN - time differential, minute component.
472 C
473 INTEGER CDAY, CHOUR, CMIN, CSEC, DDAY, DHOUR, DMIN
474 C
475 CDAY = DAY
476 CHOUR = HOUR
477 CMIN = MIN
478 CSEC = SEC
479 C
480 DSEC = CSEC - MSEC
481 IF(DSEC.LT.0)THEN
482 DSEC = DSEC + 60
483 CMIN = CMIN - 1
484 ENDIF
485 C
486 DMIN = CMIN - MMIN
487 IF(DMIN.LT.0)THEN
488 DMIN = DMIN + 60
489 CHOUR = CHOUR - 1
490 ENDIF
491 C
492 DHOUR = CHOUR - MHOUR
493 IF(DHOUR.LT.0)THEN
494 DHOUR = DHOUR + 24
495 CDAY = CDAY - 1
496 ENDIF
497 DDAY = CDAY - MDAY
498 C
499 DSEC = DSEC + (60*DMIN) + (3600*DHOUR) + (86400*DDAY)
500 C
501 RETURN
502 END
503 C=====
504 SUBROUTINE UPDATE
505 C=====
506 C This routine is intended to emulate a real time I/O task which is
507 C connected to hardware I/O channels. Instead, it receives input from
508 C a 'boundary value' file to define changes in sensor readings.
509 C The following sensors and actuators are supported:
510 C EMCS DIGITAL OUTPUTS VARIABLE NAME 0 1
511 C Supply Fan On/Off ON off on
512 C G.A. Damper Off/Auto ECON off auto
513 C Ventilation Damper Open/Close VENT open close
514 C Zone Thermostat Setback On/Off SETBAK off on
515 C
516 C EMCS ANALOG OUTPUTS VARIABLE NAME
517 C Control Point Adjustment, Supply Air Temperature (F) SUPPLY
518 C Control Point Adjustment, Zone Temperature (F) ZONE
519 C
520 C EMCS ANALOG INPUTS VARIABLE NAME
521 C Temperature, Outside Air (F) TOUTA

```

```

522 C Temperature, Mixed Air (F) TMIXA
523 C Temperature, Supply Air (F) TSUPA
524 C Temperature, Return Air (F) TRET
525 C Temperature, Space (F) TZONEA
526 C Relative Humidity, Outside Air (%) RHOUTA
527 C Relative Humidity, Return Air (%) RHRETA
528 C Relative Humidity, Space (%) RHZONA
529 C Position, Reheat Valve (Btu/s) ZHCDEM
530 C
531 C EMCS DIGITAL INPUTS
532 C Differential Pressure Switch, Supply Fan DPSSF
533 C
534 C Input variables: none
535 C Output variables: none
536 C Procedures called: RELHUM
537 C Common blocks: UNITS, TIME, ASENSR, COMMND, DSENSR, CPA, STATE1
538 C
539 C INTEGER CLU, FLU, ILU, PLU
540 C COMMON / UNITS / CLU, FLU, ILU, PLU
541 C
542 C INTEGER DAY, HOUR, MINUTE, SECOND, ENDDAY, ENDHR, ENDMIN, ENDSEC
543 C COMMON / TIME / DAY, HOUR, MINUTE, SECOND, ENDDAY, ENDHR, ENDMIN, ENDSEC
544 C
545 C REAL TOUTA, RHOUTA, TZONEA, RHZONA, TRET,
546 C & COMMON / ASENSR / RHRETA, ZHCDEM, TSUPA, TMIXA
547 C & COMMON / ASENSR / TOUTA, RHOUTA, TZONEA(5), RHZONA(5), TRET(5),
548 C & COMMON / ASENSR / RHRETA(5), ZHCDEM(5), TSUPA(5), TMIXA(5)
549 C
550 C LOGICAL ON, ECON, VENT, SETBAK
551 C COMMON / COMMND / ON(5), ECON(5), VENT(5), SETBAK(5)
552 C
553 C LOGICAL DPSSF
554 C COMMON / DSENSR / DPSSF(5)
555 C
556 C REAL SUPPLY, ZONE
557 C COMMON / CPA / SUPPLY(5), ZONE(5)
558 C
559 C REAL A
560 C LOGICAL L
561 C COMMON / STATE1 / A(100), L(100)
562 C
563 C Global Variables:
564 C EVENT - Logical variable which is true when a command from the EMCS
565 C file has been received.
566 C LOGICAL EVENT
567 C GLOBAL EVENT
568 C
569 C Local Variables:
570 C ANSWER - character variable to receive answer on debug print question.
571 C DATE - day number for the next command event
572 C DBINIT - logical variable, true if debug print question has been asked.
573 C DBP - logical variable, true if debug print is enabled.
574 C EX - exponential of ST.
575 C LEFT - seconds left before next command event will take place.
576 C NBLD - building number used as index in EMCS variable arrays.
577 C ST - negative reciprocal of TCSENS.
578 C TCSENS - time constant of EMCS analog sensors in seconds.
579 C TIME - time in hours, minutes, and seconds for the next command event.

```

```

580 C WAIT - logical variable, true if command event has been read from
581 C command event file and is being stored, False if no command
582 C event is stored.
583 C
584 REAL EX,ST,TCSENS
585 LOGICAL WAIT,DBP,DBINIT
586 INTEGER DATE,TIME(3),LEFT
587 CHARACTER*1 ANSWER
588 C
589 PARAMETER (NBLD=1, TCSENS=15.0, ST=(-1.0)/TCSENS)
590 C
591 DATA DBP /.FALSE./, DBINIT/.FALSE./, WAIT/.FALSE./
592 C
593 NAMELIST /DEBUG/ DAY,HOUR,MINUTE,SECOND,TOUTA,RHOUTA,TZONEA,
594 &RHZONA,TRETA,RHRETA,ZHCDEM,TSUPA,TMIXA,DPSSF
595 C
596 -----
597 C Read the boundary value file to see if EMCS has sent a command. The
598 C command will have a time associated with it which is the time the
599 C command was given. The command data is not changed until the command
600 C time. The variable WAIT, if true, indicates that the routine is waiting
601 C for the command time to arrive before reading the boundary file again.
602 C
603 IF(.NOT.WAIT)THEN
604 500 READ(FLU,*,ERR=500,END=900) DATE,TIME,ON(NBLD),ECON(NBLD),
605 & VENT(NBLD),SETBAK(NBLD),
606 & SUPPLY(NBLD),ZONE(NBLD)
607 WAIT = .TRUE.
608 ENDIF
609 C-----
610 C Determine if the time for the command obtained from the boundary file
611 C has come. If so, then change the command data in common, and set the
612 C event flag to indicate a command has occurred. LEFT is the time left
613 C before the command becomes active.
614 C
615 900 IF(WAIT)THEN
616 CALL SBTIME(DATE,TIME(1),TIME(2),TIME(3),DAY,HOUR,MINUTE,SECOND,
617 & LEFT)
618 IF(LEFT.LE.0)THEN
619 L(1)=ON(NBLD)
620 L(2)=ECON(NBLD)
621 L(3)=VENT(NBLD)
622 L(4)=SETBAK(NBLD)
623 A(80)=SUPPLY(NBLD)
624 A(81)=ZONE(NBLD)
625 WAIT=.FALSE.
626 EVENT=.TRUE.
627 ELSE
628 EVENT=.FALSE.
629 ENDIF
630 ELSE
631 EVENT = .FALSE.
632 ENDIF
633 C-----
634 C Update the sensor data in common with the data from the state variables
635 C calculated by the emulator model. To simulate sensors with finite time
636 C constants, an exponential approach is used.
637 C

```

```

638 DPSSF(NBLD) = L(10)
639 C
640 EX=EXP(ST)
641 TOUTA = (1.-EX)*A(01) + EX*TOUTA
642 TRETA(NBLD) = (1.-EX)*A(20) + EX*TRETA(NBLD)
643 TZONEA(NBLD) = (1.-EX)*A(06) + EX*TZONEA(NBLD)
644 TSUPA(NBLD) = (1.-EX)*A(32) + EX*TSUPA(NBLD)
645 TMIXA(NBLD) = (1.-EX)*A(23) + EX*TMIXA(NBLD)
646 ZHCDEM(NBLD) = A(62)
647
648 RHOUTA = (1.-EX)*RELHUM(A(01),A(02),29.921) +EX*RHOUTA
649 RHRETA(NBLD) = (1.-EX)*RELHUM(A(20),A(22),29.921) +EX*RHRETA(NBLD)
650 RHZONA(NBLD) = (1.-EX)*RELHUM(A(06),A(07),29.921) +EX*RHZONA(NBLD)
651 C
652 IF(.NOT.DBINIT)THEN
653 WRITE(1,FMT='(1X,"DEBUG PRINTOUT?")')
654 READ(1,FMT='(A1)')ANSWER
655 IF(ANSWER.EQ.'Y') DBP = .TRUE.
656 DBINIT = .TRUE.
657 ENDIF
658 IF(DBP) WRITE(1,DEBUG)
659 RETURN
660 END
661 C Version 1.28 - FEBRUARY 13, 1985 - W.B. MAY , NATIONAL BUREAU OF STANDARDS
662 C=====
663 SUBROUTINE EMODEL
664 C=====
665 C This is the main subroutine for the simulation routines for the EMCS
666 C algorithm tester. The simulation sections consist of the weather data
667 C section, the Air Handling Unit simulation, the zone simulation, the
668 C local heating and cooling equipment simulation, the comfort
669 C simulation, and the data compilation routines. NOTE: all common block
670 C variables are described in subroutine PARAMA.
671 C
672 C Input variables: none
673 C Output variables: none
674 C Procedures called: AHU,CMPILE,COMFRT,CONPAR,ECTROL,LOCAL,PARAMA,WEATHR,
675 C ZONE
676 C Common blocks: CLIMAT,STATE1
677 C
678 REAL WDAVG,CWIND,TDBAVG,TDBAMP,SINORG,CSOLR
679 COMMON / CLIMAT / WDAVG,CWIND,TDBAVG,TDBAMP,SINORG,CSOLR
680 C
681 REAL A
682 LOGICAL L
683 COMMON / STATE1 / A(100),L(100)
684 C
685 C Local variables
686 C
687 C GO - logical variable; if true, indicates that EMODEL should be
688 C executed.
689 C NOINIT - logical variable; if true, indicates the the simulation
690 C parameters have not been initialized.
691 C TIMEST - The current value to be used for the major simulation time step.
692 C
693 LOGICAL NOINIT,GO
694 INTEGER TIMEST
695 DATA NOINIT/.TRUE./

```

```

696      C
697      C-----
698      C Initialize all non-time-dependent parameters
699          IF(NOINIT)THEN
700              CALL CONPAR
701              CALL PARAMA
702              NOINIT = .FALSE.
703              CALL WEATHR
704              CALL ECTROL(TIMEST,GO)
705          ELSE
706      C-----
707      C Determine if simulation routine is to be executed
708      C
709          CALL ECTROL(TIMEST,GO)
710          IF(.NOT.(GO)) RETURN
711      C-----
712      C Determine current weather parameters for this time step.
713      C
714          CALL WEATHR
715      C-----
716      C Simulate the air handling unit equipment given return air temperature,
717      C outdoor air conditions and the state of control signals. This is an
718      C algebraic/logical model assuming a very small time constant.
719      C
720          CALL AHU
721      C-----
722      C Simulate a building zone given the supply air conditions and
723      C environmental conditions. This requires solution of differential
724      C equations.
725      C
726          CALL ZONE(TIMEST)
727      C-----
728      C Determine the comfort level in the building zone
729      C
730          CALL COMFRT(TIMEST)
731      C-----
732      C Compile the data generated for later evaluation
733      C
734          ENDIF
735          CALL CMPILE(TIMEST)
736      C
737          RETURN
738          END
739      C=====
740          SUBROUTINE PARAMA
741      C=====
742      C This routine is used to initialize the emulator building simulation
743      C parameters. It is called once at the beginning of a one season test.
744      C
745      C Input variables: none
746      C Output variables: none
747      C Procedures called: none
748      C Common blocks: TIME,CLIMAT,HVAC,HVAC2,PFILES,SHELL,STATE1,UNITS,USE
749      C
750      C Description of Common blocks:
751      C TIME - contains the current time vector for the simulation.
752      C CLIMAT - contains parameters which describe climate around the building.
753      C HVAC - contains parameters which describe the HVAC system of the building.

```



```

754 C HVAC2 - contains character data describing the HVAC system of the building.
755 C PFILES - contains the character names of the files containing parameters.
756 C SHELL - contains parameters which describe the building shell.
757 C STATE1 - contains the state variables for the emulator model.
758 C UNITS - contains the logical unit assignments for program read and write.
759 C USE - contains parameters which describe building use schedules.
760 C
761 C Common Block Variables:
762 C
763 C DAY - Current day number, first day of test is day 1.
764 C HOUR - Current hour of day, from 0 - 23.
765 C MINUTE - Current minute within the hour, 0 - 59.
766 C SECOND - Current second within the minute, 0 - 59.
767 C ENDDAY - Day number on which the test is to stop.
768 C ENDHR - Hour of the day on ENDDAY within which test is to stop.
769 C ENDMIN - Minute within ENDHR within which test is to stop.
770 C ENDSEC - Second within ENDMIN within which test is to stop.
771 C
772 C INTEGER DAY, HOUR, MINUTE, SECOND, ENDDAY, ENDHR, ENDMIN, ENDSEC
773 COMMON / TIME / DAY, HOUR, MINUTE, SECOND, ENDDAY, ENDHR, ENDMIN, ENDSEC
774 C
775 C WQAVG - Average outdoor humidity ratio in lb water/lb dry air.
776 C CWIND - Average windspeed, MPH
777 C TDBAVG - Average outdoor dry bulb temperature, F
778 C TDBAMP - Diurnal amplitude of outdoor dry bulb temperature, F
779 C SINORG - Origin of sine wave describing dry bulb temperature variation in
780 seconds after midnight.
781 C CSOLR - Peak solar heat gain on horizontal.
782 C
783 C REAL WQAVG, CWIND, TDBAVG, TDBAMP, SINORG, CSOLR
784 COMMON / CLIMAT / WQAVG, CWIND, TDBAVG, TDBAMP, SINORG, CSOLR
785 C
786 C DUDECK - logical variable which is true for a dual deck system.
787 C HUMCON - logical variable which is true for a humidity control system.
788 C RAFAN - logical variable which is true for a Return Air Fan in system.
789 C VAV - logical variable which is true for a Variable Air Volume System.
790 C VOLSAF - Air handling unit fan rated supply air volume in ft3/sec.
791 C CAPLOC - Local equipment capacity for heating in Btu/sec.
792 C PGAINL - Local equipment controller proportional gain in Btu/sec. F
793 C TCTHSE - Local equipment controller sensor time constant in sec.
794 C QRAFAN - Return Air Fan air heating rate in Btu/sec.
795 C FVAMIN - Minimum ventilation air, fraction of supply air mass
796 C QSAFAN - Supply Air Fan air heating rate in Btu/sec.
797 C HCAP1 - Cold Deck Heating Coil Capacity in Btu/sec.
798 C CCAP1 - Cold Deck Cooling Coil Capacity in Btu/sec.
799 C DTSETB - Local equipment controller setpoint setback in F
800 C POWSAF - power requirement of supply fan at rated volume in kW
801 C POWRAF - power requirement of return fan at rated volume in kW
802 C GAINIL - Local equipment controller integral gain in Btu/s2 F
803 C LOCEQT - character description of Local equipment type
804 C
805 C LOGICAL DUDECK, HUMCON, RAFAN, VAV
806 COMMON / HVAC / DUDECK, HUMCON, RAFAN, VAV, VOLSAF, CAPLOC, PGAINL,
807 & TCTHSE, QRAFAN, FVAMIN, QSAFAN, HCAP1, CCAP1, DTSETB,
808 & POWSAF, POWRAF, GAINIL
809 CHARACTER*10 LOCEQT
810 COMMON / HVAC2/ LOCEQT
811 C

```

```

812 C HVFILE - Character filename of HVAC equipment parameter file.
813 C USFILE - Character filename of Building usage parameter file.
814 C CLFILE - Character filename of climate description parameter file.
815 C SHFILE - Character filename of building shell parameter file.
816 C INFILE - Character filename of state vector initialization file.
817 C SEASON - Character name of weather season to use in test.
818 C
819 CHARACTER*15 HVFILE,USFILE,CLFILE,SHFILE,INFILE,SEASON
820 COMMON / PFILES / HVFILE,USFILE,CLFILE,SHFILE,INFILE,SEASON
821 C
822 C AREAW - Area of the building wall exposed to the outside in ft2.
823 C HIWALL - Inner wall film coefficient in Btu/s F ft2.
824 C HOWALL - Outer wall film coefficient in Btu/s F ft2.
825 C LENOW - thickness of wall outside the mass center in ft.
826 C LENIW - thickness of wall inside the mass center in ft.
827 C KOW - wall conductivity outside the mass center in Btu/s F ft.
828 C KIW - wall conductivity inside the mass center in Btu/s F ft.
829 C CFUR - thermal capacity of zone furnishings in Btu/F.
830 C VOLAIR - volume of air in zone in ft3.
831 C CPW - specific heat of wall outside the mass center in Btu/lb F.
832 C RHOW - density of wall outside the mass center in lbm/ft3.
833 C CPIW - specific heat of wall inside the mass center in Btu/lb F.
834 C RHOIW - density of wall inside the mass center in lbm/ft3.
835 C AIREXC - base air infiltration rate without wind and thermal in ACH
836 C RMGFUR - furnishings moisture absorption mass relative to air
837 C
838 REAL AREAW,HIWALL,HOWALL,LENOW,LENIW,KOW,KIW,CFUR,
839 & VOLAIR,CPW,RHOW,CPIW,RHOIW,AIREXC,RMGFUR
840 COMMON / SHELL / AREAW,HIWALL,HOWALL,LENOW,LENIW,KOW,KIW,CFUR,
841 & VOLAIR,CPW,RHOW,CPIW,RHOIW,AIREXC,RMGFUR
842 C
843 C contents of state vector
844 C A(1): TEMPERATURE of outside air (F)
845 C A(2): HUMIDITY RATIO of outside air (lbw/lbda)
846 C A(3): ENTHALPY of outside air (Btu/lb)
847 C A(4): Windspeed (MPH)
848 C
849 C A(6): TEMPERATURE of zone air (F)
850 C A(7): HUMIDITY RATIO of zone air (lbw/lbda)
851 C A(8): TEMPERATURE of zone wall interior (F)
852 C A(9): TEMPERATURE of Zone wall surface (F)
853 C A(10): TEMPERATURE of zone glass (F)
854 C
855 C A(20): TEMPERATURE of return air (F)
856 C A(21): ENTHALPY of return air (BTU/lb)
857 C A(22): HUMIDITY RATIO of return air (lbw/lbda)
858 C A(23): TEMPERATURE of mixed air (F)
859 C A(24): ENTHALPY of mixed air (BTU/lb)
860 C A(25): HUMIDITY RATIO of mixed air (lbw/lbda)
861 C A(26): TEMPERATURE of cooling coil discharge air (F) cold deck
862 C A(27): ENTHALPY of cooling coil discharge air (BTU/lb) cold deck
863 C A(28): HUMIDITY RATIO of cooling coil discharge air (lbw/lbda) cold deck
864 C A(29): TEMPERATURE of heating coil discharge air (F) cold deck
865 C A(30): ENTHALPY of heating coil discharge air (BTU/lb) cold deck
866 C A(31): HUMIDITY RATIO of heating coil discharge air (lbw/lbda) cold deck
867 C A(32): TEMPERATURE of supply air (F) cold deck
868 C A(33): ENTHALPY of supply air (BTU/lb) cold deck
869 C A(34): HUMIDITY RATIO of supply air (lbw/lbda) cold deck

```

```

870 C A(35): ENTHALPY of return air plus ventilation air(BTU/lb)
871 C
872 C A(51): MASS FLOW RATE of return air (lb/s)
873 C A(52): MASS FLOW RATE of supply air (lb/s) cold deck
874 C A(53): MASS FLOW RATE of ventilation air (lb/s)
875 C
876 C A(60): HEATING POWER to heating coil (Btu) cold deck
877 C A(61): COOLING POWER from cooling coil (Btu) cold deck
878 C A(62): HEATING POWER to local zone equipment (Btu/s)
879 C A(63): COOLING POWER from local zone equipment (Btu/s)
880 C A(64): HEATING ENERGY (requirements) (Btu)
881 C A(65): COOLING ENERGY (requirements) (Btu)
882 C A(66): HEATING ENERGY (economizer) (Btu)
883 C A(67): COOLING ENERGY (economizer) (Btu)
884 C A(68): HEATING ENERGY (load) (Btu)
885 C A(69): COOLING ENERGY (load) (Btu)
886 C A(70): HEATING ENERGY (reheat) (Btu)
887 C A(71): COOLING ENERGY (recool) (Btu)
888 C A(72): ELECTRICAL ENERGY to fans (kWh)
889 C A(73): HEATING ENERGY to local zone equipment (Btu)
890 C A(74): COOLING ENERGY from local zone equipment (Btu)
891 C
892 C A(80): CONTROL POINT ADJUSTMENT supply air temperature (F)
893 C A(81): CONTROL POINT ADJUSTMENT zone temperature (F)
894 C A(82): CONTROL POINT ADJUSTMENT outside air damper position (F)
895 C A(83): CONTROL POINT ADJUSTMENT supply air RH (%)
896 C
897 C A(90): COMFORT seconds outside of comfort range
898 C A(91): COMFORT maximum dry bulb temperature (F)
899 C A(92): COMFORT minimum dry bulb temperature (F)
900 C A(93): COMFORT relative humidity at maximum temperature (%)
901 C A(94): COMFORT relative humidity at minimum temperature (%)
902 C
903 C L(01): COMMAND air handling unit on/off
904 C L(02): COMMAND economizer on/off
905 C L(03): COMMAND ventilation on/off
906 C L(04): COMMAND zone setback on/off
907 C
908 C L(10): STATUS air handling unit on/off
909 C L(11): STATUS zone occupied/unoccupied
910 C
911 REAL A
912 LOGICAL L
913 COMMON / STATE1 / A(100),L(100)
914 C
915 C CLU - Logical Unit for writes to console.
916 C FLU - Logical Unit for read from EMCS command input file.
917 C ILU - Logical Unit for read from console.
918 C PLU - Logical Unit for read from parameter files.
919 C
920 INTEGER CLU,FLU,ILU,PLU
921 COMMON / UNITS / CLU,FLU,ILU,PLU
922 C
923 C LSCHED - Index of current building usage state description in arrays.
924 C NSCHED - Number of possible states in schedule arrays.
925 C TSCHED - Start time of LSCHED'th building usage state in hours past
926 C midnight.
927 C SENSIB - Equipment sensible heat gain for LSCHED'th building usage state

```

```

928 C          In Btu/s.
929 C LATENT - Equipment moisture gain for LSCHEd'th building usage state
930 C          In lbm/s.
931 C PESENS - Sensible heat gain from people for LSCHEd'th building usage state
932 C          In Btu/s.
933 C PELATE - Moisture gain from people for LSCHEd'th building usage state
934 C          In lbm/s
935 C
936 C          PARAMETER (NSCHED = 10)
937 C          INTEGER          LSCHEd
938 C          REAL              TSCHED          ,SENSIB,PESENS, LATENT,PELATE
939 C          COMMON / USE / LSCHEd,TSCHED(NSCHED),SENSIB(NSCHED),LATENT(NSCHED)
940 C          &                  ,PESENS(NSCHED),PELATE(NSCHED)
941 C
942 C Local Variables:
943 C AINIT - Temporary value to hold state vector initializer read from
944 C          initialization file.
945 C BTUSPW - Conversion factor, Btu per Watt.
946 C BULB - Indicates type of lighting in describing building schedule.
947 C CHMAX - Time of maximum outdoor dry bulb temperature, hrs from midnight.
948 C CTMAX - Maximum daily outdoor temperature for a particular season in F.
949 C CTMIN - Minimum daily outdoor temperature for a particular season in F.
950 C EQUIPL - Equipment moisture gain for a schedule period in lb water/hr.
951 C EQUIPS - Electric power used by any equipment in zone for a schedule period
952 C          in Watts.
953 C INDEX - Temporary value to hold state vector index read from initialization
954 C          file.
955 C LIGHT - Lighting power in the zone for a schedule period in watts.
956 C LINIT - Temporary variable to hold logical state vector initializer read
957 C          from initialization file.
958 C NACCAT - Number of possible activity categories for occupants of zone.
959 C NACTIV - Array containing number of occupants in NACCAT activity categories
960 C          for a schedule period.
961 C OCCLAT - Occupant moisture gain per person for a schedule period in
962 C          lb water/hour.
963 C OCCSEN - Occupant sensible heat gain per person for a schedule period in
964 C          Btu/hr.
965 C RE - ratio of radiated energy to total equipment energy for a schedule
966 C          period.
967 C RL - ratio of radiated energy to total lighting energy for a given
968 C          lighting type.
969 C RP - ratio of radiated energy to total sensible heat of people.
970 C SEASIN - temporary variable to hold season name from climate file for
971 C          comparison with desired season in variable SEASON.
972 C TOLAT - total moisture gain from occupants for a schedule period in lb.
973 C          water/hour.
974 C TOSEN - total sensible heat gain of occupants for a schedule period in
975 C          Btu/hour.
976 C
977 C          REAL EQUIPL,EQUIPS,RE,RL,RP,TOLAT,TOSEN
978 C          PARAMETER (NACCAT = 5,BTUSPW = 9.47817E-4, RP = 0.4)
979 C          LOGICAL LINIT
980 C          INTEGER NACTIV(NACCAT),INDEX
981 C          REAL AINIT,CHMAX,CTMAX,CTMIN,OCCSEN(NACCAT),OCCLAT(NACCAT),LIGHT
982 C          CHARACTER*1 BULB
983 C          CHARACTER*5 SEASIN
984 C
985 C          DATA OCCSEN / 230., 255., 315., 345., 345. /

```

```

986 DATA CCCLAT / 0.20, 0.27, 0.34, 0.46, 0.74 /
987 C
988 C      Occupant sensible and moisture gains:
989 C      (from ASHRAE fundamentals, 1981, p 26.25 and Threlkeld, page 363)
990 C      sensible BTU/hr  moisture LB/hr
991 C  1. seated, very light work          230.          0.20
992 C  2. seated, light work, typing      255.          0.27
993 C  3. standing, light work            315.          0.34
994 C  4. light bench work                 345.          0.46
995 C  5. walking, 3 mph, light machine work 345.          0.74
996 C
997 C-----
998 C Initialize State Variables
999 C
1000 OPEN(PLU, FILE=INFILE, STATUS='OLD', ERR=994)
1001 DO 250 I = 1,100
1002   READ(PLU, *, END=280, ERR=260) INDEX, AINIT
1003   A(INDEX) = AINIT
1004 250 CONTINUE
1005 260 DO 270 J = 1,100
1006   READ(PLU, *, END=280, ERR=993) INDEX, LINIT
1007   L(INDEX) = LINIT
1008 270 CONTINUE
1009 280 CLOSE(PLU)
1010 C-----
1011 C Read Building Use parameters
1012 C
1013 OPEN(PLU, FILE=USFILE, STATUS='OLD', ERR=998)
1014 LSCHED = 0
1015 READ(PLU, *, END=500)
1016 DO 400 I = 1, NSCHED
1017 C
1018 C TSCHED in hours, LIGHT and EQUIPS in W, EQUIPL in lb water/hr
1019 C
1020 READ(PLU, *, END=500) TSCHED(I), LIGHT, BULB, EQUIPS, RE, EQUIPL,
1021 & (NACTIV(K), K=1, NACCAT)
1022 C
1023 C Determine lighting type, I = incandescent, F = fluorescent
1024 C
1025 IF(BULB.EQ.'I') THEN
1026   RL = 0.8
1027 ELSE IF(BULB.EQ.'F') THEN
1028   RL = 0.5
1029 ELSE
1030   RL = 0.5
1031 ENDIF
1032 C
1033 LSCHED = LSCHED + 1
1034 TOSEN = 0.0
1035 TOLAT = 0.0
1036 DO 300 J = 1, NACCAT
1037   TOSEN = TOSEN + NACTIV(J) * OCCSEN(J)
1038   TOLAT = TOLAT + NACTIV(J) * CCCLAT(J)
1039 300 CONTINUE
1040 SENSIB(I) = ((1-RL)*LIGHT + (1-RE)*EQUIPS) * BTUSPW
1041 LATENT(I) = EQUIPL / 3600.
1042 PESENS(I) = (1-RP)*TOSEN / 3600.
1043 PELATE(I) = TOLAT / 3600.

```

```

1044      400 CONTINUE
1045      C
1046      C-----
1047      C Read Building Shell Parameters
1048      C
1049      500 OPEN(PLU, FILE=SHFILE, STATUS='OLD', ERR=999)
1050      READ(PLU, *) AREAW
1051      READ(PLU, *) HIWALL
1052      READ(PLU, *) HOWALL
1053      READ(PLU, *) LENGW
1054      READ(PLU, *) LENIW
1055      READ(PLU, *) KGW
1056      READ(PLU, *) KIW
1057      READ(PLU, *) CFUR
1058      READ(PLU, *) VOLAIR
1059      READ(PLU, *) CPOW
1060      READ(PLU, *) RH00W
1061      READ(PLU, *) CPIW
1062      READ(PLU, *) RH0IW
1063      READ(PLU, *) AIREXC
1064      READ(PLU, *) RMGFUR
1065      CLOSE(PLU)
1066      C
1067      C-----
1068      C Read HVAC system Parameters
1069      C
1070      OPEN(PLU, FILE=HVFILE, STATUS='OLD', ERR=997)
1071      READ(PLU, *) VOLSAF
1072      READ(PLU, *) POWSAF
1073      READ(PLU, *) POWRAF
1074      READ(PLU, *) LOCEQT
1075      READ(PLU, *) CAPLOC
1076      READ(PLU, *) PGAINL
1077      READ(PLU, *) GAINIL
1078      READ(PLU, *) TCTHSE
1079      READ(PLU, *) VAV
1080      READ(PLU, *) RAFAN
1081      READ(PLU, *) GRAFAN
1082      READ(PLU, *) FVAMIN
1083      READ(PLU, *) DUDECK
1084      READ(PLU, *) QSAFAN
1085      READ(PLU, *) HUMCON
1086      READ(PLU, *) HCAP1
1087      READ(PLU, *) CCAP1
1088      READ(PLU, *) DTSETB
1089      CLOSE(PLU)
1090      C-----
1091      C Read Climate Parameters
1092      C
1093      OPEN(PLU, FILE=CLFILE, STATUS='OLD', ERR=996)
1094      READ(PLU, *) SEASIN
1095      600 READ(PLU, *, END=995) SEASIN
1096      READ(PLU, *) CTMAX
1097      READ(PLU, *) CTMIN
1098      READ(PLU, *) CHMAX
1099      READ(PLU, *) W0AVG
1100      READ(PLU, *) CS0LR
1101      READ(PLU, *) CWIND

```

```

1102     IF(SEASIN(1:5).NE.SEASON(1:5)) GO TO 600
1103     CLOSE(PLU)
1104     TDBAVG = (CTMAX + CTMIN) /2.
1105     TDBAMP = CTMAX - TDBAVG
1106     SINORG = (CHMAX-30.)*3600.
1107
1108     C
1109     RETURN
1110     C-----ERRORS-----
1111     993 WRITE(CLU,7)I+J
1112     7 FORMAT(1X,'unable to read INITIALIZATION file at line',I6)
1113     STOP
1114     994 WRITE(CLU,6)
1115     6 FORMAT(1X,'unable to open INITIALIZATION file')
1116     STOP
1117     995 WRITE(CLU,5)
1118     5 FORMAT(1X,'unable to locate requested season in CLIMATE file')
1119     STOP
1120     996 WRITE(CLU,4)
1121     4 FORMAT(1X,'unable to open CLIMATE parameter file')
1122     STOP
1123     997 WRITE(CLU,3)
1124     3 FORMAT(1X,'unable to open HVAC parameter file')
1125     STOP
1126     998 WRITE(CLU,2)
1127     2 FORMAT(1X,'unable to open USE parameter file')
1128     STOP
1129     999 WRITE(CLU,1)
1130     1 FORMAT(1X,'unable to open SHELL parameter file')
1131     STOP
1132     END
1133     C=====
1134     SUBROUTINE WEATHR
1135     C=====
1136     C This routine is used to determine the current weather conditions. Wind
1137     C and humidity ratio are considered constants. Outdoor temperature is
1138     C assumed to vary sinusoidally with a period of 24 hours.
1139     C
1140     C AS of 2-15-85 there is no solar energy section.
1141     C
1142     C Input variables: none
1143     C Output variables: none
1144     C Procedures called: SINUSG
1145     C Common blocks: STATE1, CLIMAT, TIME
1146     C
1147     REAL          A
1148     LOGICAL       L
1149     COMMON / STATE1 / A(100),L(100)
1150     C
1151     INTEGER       DAY, HOUR, MINUTE, SECOND, ENDDAY, ENDHR, ENDMIN, ENDSEC
1152     COMMON / TIME / DAY, HOUR, MINUTE, SECOND, ENDDAY, ENDHR, ENDMIN, ENDSEC
1153     C
1154     COMMON / CLIMAT / WCAVG, CWIND, TDBAVG, TDBAMP, SINORG, CSOLR
1155     C
1156     C Local Variables:
1157     C
1158     C PERIOD - Period of sinusoidal temperature variation in seconds.
1159     C RTIME - Current time in seconds past midnight.
1160     C TOUTA - Outside Air dry bulb temperature in F

```

```

1160 C VWIND - Wind velocity in MPH.
1161 C WOUTA - Outside air humidity ratio in lb water / lb dry air
1162 C
1163 REAL PERIOD,RTIME
1164 PARAMETER (PERIOD=86400.)
1165 EQUIVALENCE ( A(1),TOUTA ),( A(2),WOUTA ),(A(04),VWIND)
1166 C
1167 RTIME = HOUR*3600 + MINUTE*60 + SECOND
1168 CALL SINUSO(TDBAVG,PERIOD,TDBAMP,RTIME,SINORG,TOUTA)
1169 WOUTA = WDAVG
1170 VWIND = CWIND
1171 RETURN
1172 END
1173 C=====
1174 SUBROUTINE SINUSO(TDBAVG,PERIOD,AMP,RTIME,RORIG,T)
1175 C=====
1176 C Returns a point on an approximate sinusoidal curve as a function of
1177 C time. The curve shape is based on table data in SHAPE.
1178
1179 C Input variables:
1180 C TDBAVG - the average value or baseline of the curve.
1181 C PERIOD - contains the period of the wave in seconds.
1182 C AMP - the amplitude of the curve with reference to the baseline.
1183 C RTIME - the current time in seconds since midnight.
1184 C RORIG - the origin of the wave in seconds since midnite.
1185
1186 C Output variables:
1187 C T - the current value of the wave at time RTIME.
1188 C
1189 C Procedures called: none
1190 C Common blocks: none
1191 C
1192 C Local Variables:
1193 C
1194 C ICURP2 - Index into SHAPE table for smallest time larger than current time.
1195 C ICURPE - Index into SHAPE table for largest time smaller than current time.
1196 C NPER - Number of whole curve periods passed by current time.
1197 C NPIECE - Number of entries in curve look-up table.
1198 C RANGE - Difference between two table entries for times on either side
1199 C of current time.
1200 C RCURPE - Real number of curve pieces past origin for current time.
1201 C RNPPS - Real number of curve pieces per second.
1202 C SHAPE - Curve look-up table containing normalized curve values for one
1203 C period of oscillation equally divided into NPIECE parts.
1204 C
1205 C PARAMETER (NPIECE = 24)
1206 C REAL SHAPE(NPIECE)
1207 C DATA SHAPE/+0.0000,+0.2588,+0.5000,+0.7071,+0.8660,+0.9659,
1208 C & +1.0000,+0.9659,+0.8660,+0.7071,+0.5000,+0.2588,
1209 C & -0.0000,-0.2588,-0.5000,-0.7071,-0.8660,-0.9659,
1210 C & -1.0000,-0.9659,-0.8660,-0.7071,-0.5000,-0.2588/
1211 C
1212 C RNPPS = FLOAT(NPIECE) / PERIOD
1213 C RCURPE = RNPPS * (RTIME - RORIG)
1214 C ICURPE = AINT (RCURPE)
1215 C NPER = ICURPE / NPIECE
1216 C ICURPE = ICURPE - NPER * (NPIECE)
1217 C RCURPE = RCURPE - NPER * (NPIECE)

```



```

1218      ICURP2 = ICURPE + 2
1219      IF(ICURP2.GT.NPIECE) ICURP2 = ICURP2 - NPIECE
1220      RANGE = (SHAPE(ICURP2) - SHAPE(ICURPE+1))
1221      T = ((RCURPE - ICURPE) * RANGE + SHAPE(ICURPE+1)) * AMP + TDBAVG
1222      RETURN
1223      END
1224  C=====
1225      SUBROUTINE AHU
1226  C=====
1227  C This routine contains a steady state model of a generalized air handling
1228  C unit.
1229  C Input variables: none
1230  C Output variables: none
1231  C Procedures called: CPAIR, DEWPT, ENTHAL, HUMRAT, RHOAIR, TDBSAT, TEMP
1232  C Common blocks: HVAC, HVAC2, STATE1
1233  C
1234      LOGICAL          DUDECK, HUMCON, RAFAN, VAV
1235      COMMON / HVAC / DUDECK, HUMCON, RAFAN, VAV, VOLSAF, CAPLOC, PGAINL,
1236      &                TCTHSE, GRAFAN, FVAMIN, QSAFAN, HCAP1, CCAP1, DTSETB,
1237      &                POWSAF, POWRAF, GAINIL
1238      CHARACTER*10     LOCEQT
1239      COMMON / HVAC2/ LOCEQT
1240  C
1241      REAL              A
1242      LOGICAL           L
1243      COMMON / STATE1 / A(100),L(100)
1244  C
1245  C Local Variables:
1246  C
1247  C AHSTAT - STATUS air handling unit on/off
1248  C DELTA - Difference between supply air setpoint and fan entering air T.
1249  C DTRF - Temperature differential across the return fan.
1250  C DTSF - Temperature differential across the supply fan.
1251  C ECONO - COMMAND economizer on/off
1252  C HCA1 - ENTHALPY of cooling coil discharge air (BTU/lb) cold deck
1253  C HHA1 - ENTHALPY of heating coil discharge air (BTU/lb) cold deck
1254  C HMA - ENTHALPY of mixed air (BTU/lb)
1255  C HMC - ENTHALPY of return air plus ventilation air (BTU/lb)
1256  C HMO - Enthalpy of mixed air if outside air dampers are open (Btu/lb)
1257  C HOUTA - ENTHALPY of outside air (Btu/lb)
1258  C HRA - ENTHALPY of return air (BTU/lb)
1259  C HSA1 - ENTHALPY of supply air (BTU/lb) cold deck
1260  C MRA - MASS FLOW RATE of return air (lb/s)
1261  C MSA1 - MASS FLOW RATE of supply air (lb/s) cold deck
1262  C PRESS - Atmospheric pressure (assumed constant) in inches of Hg.
1263  C QCOOL1 - COOLING POWER from cooling coil (Btu) cold deck
1264  C QHEAT1 - HEATING POWER to heating coil (Btu) cold deck
1265  C TCA1 - TEMPERATURE of cooling coil discharge air (F) cold deck
1266  C TDPMA - Dewpoint temperature of mixed air (F)
1267  C THA1 - TEMPERATURE of heating coil discharge air (F) cold deck
1268  C TMA - TEMPERATURE of mixed air (F)
1269  C TMC - Temperature of return air plus ventilation air (F)
1270  C TMO - Temperature of mixed air if outside air dampers are open (F)
1271  C TOUTA - TEMPERATURE of outside air (F)
1272  C TRA - TEMPERATURE of return air (F)
1273  C TSA1 - TEMPERATURE of supply air (F) cold deck
1274  C TSET - CONTROL POINT ADJUSTMENT supply air temperature (F)
1275  C TZA - TEMPERATURE of zone air (F)

```

```

1276 C UNITON - COMMAND air handling unit on/off
1277 C VENOFF - COMMAND ventilation on/off
1278 C WCA1 - HUMIDITY RATIO of cooling coil discharge air (lbw/lbda) cold deck
1279 C WHA1 - HUMIDITY RATIO of heating coil discharge air (lbw/lbda) cold deck
1280 C WMA - HUMIDITY RATIO of mixed air (lbw/lbda)
1281 C WMC - Humidity ratio of return air plus ventilation air (lbw/lbda)
1282 C WMO - Humidity ratio of mixed air if outside air dampers are open (lbw/lbda)
1283 C WOUTA - HUMIDITY RATIO of outside air (lbw/lbda)
1284 C WRA - HUMIDITY RATIO of return air (lbw/lbda)
1285 C WSA1 - HUMIDITY RATIO of supply air (lbw/lbda) cold deck
1286 C WZA - HUMIDITY RATIO of zone air (lbw/lbda)
1287 C
1288 REAL MRA,MSA1
1289 EQUIVALENCE ( A(01),TOUTA ),( A(02),WOUTA ),( A(20),TRA )
1290 EQUIVALENCE ( A(22),WRA ),( A(80),TSET ),( A(24),HMA )
1291 EQUIVALENCE ( A(06),TZA ),( A(07),WZA ),( A(33),HSA1 )
1292 EQUIVALENCE ( A(32),TSA1 ),( A(34),WSA1 ),( A(52),MSA1 )
1293 EQUIVALENCE ( A(21),HRA ),( A(51),MRA ),( A(25),WMA )
1294 EQUIVALENCE ( A(26),TCA1 ),( A(28),WCA1 ),( A(27),HCA1 )
1295 EQUIVALENCE ( A(29),THA1 ),( A(31),WHA1 ),( A(30),HHA1 )
1296 EQUIVALENCE ( A(03),HOUTA ),( A(23),TMA ),( A(35),HMC )
1297 EQUIVALENCE ( A(60),QHEAT1 ),( A(61),QCOOL1 )
1298 C
1299 LOGICAL UNITON,VENOFF,ECONG,AHSTAT
1300 EQUIVALENCE ( L(01),UNITON ),( L(03),VENOFF ),( L(02),ECONG )
1301 EQUIVALENCE ( L(10),AHSTAT )
1302 C
1303 PARAMETER (PRESS = 29.921)
1304 C
1305 -----
1306 C Air handling unit is operating - determine air flow rates
1307 C
1308 IF(UNITON) THEN
1309 AHSTAT = .TRUE.
1310 HOUTA = ENTHAL(TOUTA,WOUTA,PRESS)
1311 IF(VAV) THEN
1312 C*****
1313 STOP 'NO VAV IMPLEMENTED'
1314 C*****
1315 ELSE
1316 MSA1 = VOLSAF * RHOAIR(WSA1,TSET)
1317 C*****
1318 MRA = MSA1
1319 C*****
1320 ENDIF
1321 C
1322 -----
1323 C Determine heating of air by return fan
1324 C
1325 IF(RAFAN) THEN
1326 DTRF = GRAFAN / (MRA * CPAIR(WZA) )
1327 TRA = TZA + DTRF
1328 ELSE
1329 TRA = TZA
1330 ENDIF
1331 WRA = WZA
1332 HRA = ENTHAL(TRA,WRA,PRESS)
1333 C

```

```

1334 C-----
1335 C Determine mixed air conditions
1336 C
1337 C-----state of air for outside air dampers closed-----
1338 C-----minimum outside air dampers closed-----
1339     IF(VENOFF) THEN
1340         HMC = HRA
1341         WMC = WRA
1342         TMC = TRA
1343 C-----minimum outside air dampers open-----
1344     ELSE
1345         HMC = (FVAMIN * HOUTA) + (1-FVAMIN)*HRA
1346         WMC = (FVAMIN * WOUTA) + (1-FVAMIN)*WRA
1347         TMC = TEMP(HMC,WMC,PRESS)
1348     ENDIF
1349 C-----determine mixed air temperature-----
1350     IF(.NOT.DUDECK)THEN
1351 C-----single deck-----
1352 C-----supply fan temperature rise-----
1353         DTSF = QSAFAN / (MSA1 * CPAIR(WRA))
1354         DELTA = TSET - DTSF
1355         IF(.NOT.HUMCON)THEN
1356 C-----economizer enabled-determine damper position-----
1357             IF(ECONO) THEN
1358                 HMO = HOUTA
1359                 TMO = TOUTA
1360                 WMO = WOUTA
1361 C-----outside air damper should be closed-----
1362                 IF(TMC.LE.DELTA) THEN
1363                     TMA = TMC
1364                     WMA = WMC
1365                     HMA = HMC
1366 C-----outside air damper should be open-----
1367                 ELSE IF(TMC.GT.DELTA.AND.TMO.GT.DELTA) THEN
1368                     TMA = TMO
1369                     WMA = WMO
1370                     HMA = HMO
1371 C-----outside air damper should be part open-----
1372                 ELSE IF(TMC.GT.DELTA.AND.TMO.LE.DELTA) THEN
1373                     WMA = (WMC - WMO)/(TMC - TMO)*(DELTA-TMO) + WMO
1374                     TMA = DELTA
1375                     HMA = ENTHAL(TMA,WMA,PRESS)
1376                 ENDIF
1377             ELSE
1378                 HMA = HMC
1379                 WMA = WMC
1380                 TMA = TMC
1381             ENDIF
1382 C*****C
1383 C damper position control or report could be installed C
1384 C*****C
1385 C-----
1386 C Determine heating or cooling required
1387 C
1388 C-----no heating or cooling required-----
1389     IF(TMA.EQ.DELTA) THEN
1390         TCA1 = TMA
1391         WCA1 = WMA

```

```

1392         HCA1 = HMA
1393         THA1 = TMA
1394         WHA1 = WMA
1395         HHA1 = HMA
1396         QHEAT1 = 0.0
1397         QCOOL1 = 0.0
1398 C-----heating required-----
1399         ELSE IF(TMA.LT.DELTA) THEN
1400             THA1 = DELTA
1401             WHA1 = WMA
1402             HHA1 = ENTHAL(THA1,WHA1,PRESS)
1403             QHEAT1 = MSA1 * (HHA1 - HMA)
1404             IF(QHEAT1.GT.HCAP1) THEN
1405                 HHA1 = HCAP1/MSA1 +HMA
1406                 THA1 = TEMP(HHA1,WHA1,PRESS)
1407                 QHEAT1 = HCAP1
1408             ENDIF
1409             HCA1 = HHA1
1410             TCA1 = THA1
1411             WCA1 = WHA1
1412             QCOOL1 = 0.0
1413 C----- cooling required-----
1414         ELSE IF(TMA.GT.DELTA) THEN
1415             THA1 = TMA
1416             WHA1 = WMA
1417             HHA1 = HMA
1418             QHEAT1 = 0.0
1419             TCA1 = DELTA
1420             TDPMA = DEWPT(WMA,PRESS)
1421             IF(TCA1.GT.TDPMA) THEN
1422                 HCA1 = ENTHAL(TCA1,WMA,PRESS)
1423                 WCA1 = WMA
1424             ELSE
1425                 WCA1 = HUMRAT(TCA1,100.0,PRESS)
1426                 HCA1 = ENTHAL(TCA1,WCA1,PRESS)
1427             ENDIF
1428             QCOOL1 = MSA1 * (HMA - HCA1)
1429             IF(QCOOL1.GT.CCAP1) THEN
1430                 HCA1 = HMA - CCAP1/MSA1
1431                 QCOOL1 = CCAP1
1432                 IF(HCA1.GT.ENTHAL(TDPMA,WMA,PRESS)) THEN
1433                     TCA1 = TEMP(HCA1,WMA,PRESS)
1434                     WCA1 = WMA
1435                 ELSE
1436                     TCA1 = TDBSAT(HCA1,PRESS)
1437                     WCA1 = HUMRAT(TCA1,100.0,PRESS)
1438                 ENDIF
1439             ENDIF
1440         ENDIF
1441 C-----determine supply air temperatures-----
1442         WSA1 = WCA1
1443         TSA1 = TCA1 + DTSF
1444         HSA1 = ENTHAL(TSA1,WSA1,PRESS)
1445     ELSE
1446         STOP 'HUMIDITY CONTROL NOT IMPLEMENTED'
1447     ENDIF
1448 ELSE
1449     STOP 'DUAL DECK NOT IMPLEMENTED'

```

```

1450         ENDIF
1451     C
1452 C-----
1453 C Air Handling Unit Off
1454 C
1455     ELSE
1456         AHSTAT = .FALSE.
1457         MSA1 = 0.
1458         MRA = MSA1
1459         TSA1 = TZA
1460         TRA = TZA
1461         TMA = TZA
1462         WSA1 = WZA
1463         WRA = WZA
1464         WMA = WZA
1465     ENDIF
1466     RETURN
1467     END
1468 C=====
1469     SUBROUTINE LOCAL(TIMEST)
1470 C=====
1471 C This routine contains a simple model of the local heating and/or
1472 C cooling equipment in the zone. It is assumed that the change in output
1473 C of the local equipment is proportional to the change in error between
1474 C the local controller setpoint and the room air temperature and also is
1475 C proportional to the integral of the error (error*timestep). Local
1476 C equipment may be of several types as configured with the variables in
1477 C common blocks HVAC and HVAC2.
1478 C
1479 C Input variables:
1480 C TIMEST - The current value to be used for the simulation time step.
1481 C
1482 C Output variables: none
1483 C Procedures called: ENTHAL
1484 C Common blocks: HVAC,HVAC2,STATE1
1485 C
1486     LOGICAL          DUDECK,HUMCON,RAFAN,VAV
1487     COMMON / HVAC / DUDECK,HUMCON,RAFAN,VAV,VOLSAF,CAPLOC,PGAINL,
1488     &                TCTHSE,GRAFAN,FVAMIN,QSAFAN,HCAPI,CCAPI,DTSETB,
1489     &                POWSAF,POWRAF,GAINIL
1490     CHARACTER*10     LOCEQT
1491     COMMON / HVAC2/ LOCEQT
1492 C
1493     REAL              A
1494     LOGICAL           L
1495     COMMON / STATE1 / A(100),L(100)
1496 C
1497 C Local Variables:
1498 C
1499 C DQAIR - Required change in heat contained in supply air entering zone to
1500 C        reduce zone temperature error. (Btu/s)
1501 C DQENT - Required change in total supplied heat entering zone to reduce
1502 C        zone temperature error. Equals DQAIR + DQLOC.(Btu/s)
1503 C DQLOC - Required change in auxilliary heat added to supply air by local
1504 C        zone equipment to reduce zone temperature error.(Btu/s)
1505 C ELOCC - COOLING ENERGY from local zone equipment (Btu)
1506 C ELOCH - HEATING ENERGY to local zone equipment (Btu)
1507 C ERROR - difference between local zone setpoint and zone temperature at

```

```

1508 C current time.
1509 C ERROR0 - difference between local zone setpoint and zone temperature at
1510 C previous timestep.
1511 C EX - Transition ratio for local controller temperature sensor
1512 C HSA1 - ENTHALPY of supply air (BTU/lb) cold deck
1513 C HSA2 - Enthalpy of hot deck (if existing) supply air (Btu/lb)
1514 C HZA - Enthalpy of zone air (Btu/lb)
1515 C MRA - MASS FLOW RATE of return air (lb/s)
1516 C MSA1 - MASS FLOW RATE of supply air (lb/s) cold deck
1517 C MSA2 - Mass flow rate of hot deck (if existing) supply air (lb/s)
1518 C PRESS - Atmospheric pressure (assumed constant) in inches of Hg.
1519 C QAIR - Heat energy entering zone space through supply air.
1520 C QLOCH - HEATING POWER to local zone equipment (Btu/s)
1521 C SETBAK - COMMAND zone setback on/off
1522 C TCTHSE - Time constant of temperature sensor in local controller (s).
1523 C TREF - Current local zone setpoint temperature (F).
1524 C TSETLO - CONTROL POINT ADJUSTMENT zone temperature (F)
1525 C TTHSE - Temperature measured by temperature sensor in local controller (s).
1526 C TZA - TEMPERATURE of zone air (F)
1527 C WZA - HUMIDITY RATIO of zone air (lbw/lbda)
1528 C
1529 C REAL MRA,MSA1,MSA2
1530 C LOGICAL SETBAK
1531 C EQUIVALENCE ( A(81),TSETLO ),( L(04),SETBAK ),( A(52),MSA1 )
1532 C EQUIVALENCE ( A(06),TZA ),( A(07),WZA ),( A(33),HSA1 )
1533 C EQUIVALENCE ( A(31),MRA ),( A(62),QLOCH )
1534 C EQUIVALENCE ( A(73),ELOCH ),( A(74),ELOCC )
1535 C
1536 C PARAMETER (PRESS = 29.921)
1537 C DATA TTHSE/75./,ERROR0/0./
1538 C
1539 C IF(SETBAK)THEN
1540 C TREF = TSETLO - DTSETB
1541 C ELSE
1542 C TREF = TSETLO
1543 C ENDIF
1544 C EX = EXP(-TIMEST/TCTHSE)
1545 C TTHSE = (1-EX)*TZA + EX*TTHSE
1546 C ERROR = TREF - TTHSE
1547 C DQENT = PGAINL * (ERROR-ERROR0) + GAINIL * ERROR * TIMEST
1548 C ERROR0 = ERROR
1549 C HZA = ENTHAL(TZA,WZA,PRESS)
1550 C THE FOLLOWING VALUES ARE FOR DUAL DECK (NOT IMPLEMENTED)
1551 C MSA2 = 0.0
1552 C HSA2 = 0.0
1553 C
1554 C QAIR = (HSA1 * MSA1) + (HSA2 * MSA2) - (HZA * MRA)
1555 C
1556 C---dual deck-----
1557 C IF(LOCEQT.EQ.'DUAL DECK') THEN
1558 C STOP 'DUAL DECK NOT IMPLEMENTED'
1559 C---VAV-----
1560 C ELSE IF(VAV) THEN
1561 C DQAIR = DQENT
1562 C QAIR = QAIR + DQAIR
1563 C MSA1 = QAIR / (HSA1 - HZA)
1564 C PAUSE 'ZONE VAV BOX AIR FLOW LIMITS NOT IMPLEMENTED'
1565 C---constant volume reheat-----

```

```

1566     ELSE IF(LOCEQT.EQ.'REHEAT') THEN
1567         DQLOC = DQENT
1568         QLOCH = QLOCH + DQLOC
1569         IF(QLOCH.GT.CAPLOC) QLOCH = CAPLOC
1570         IF(QLOCH.LE.0.) QLOCH = 0.
1571         ELOCH = ELOCH + QLOCH * TIMEST
1572     ELSE
1573         STOP 'LOCAL EQUIPMENT TYPE NOT IMPLEMENTED'
1574     ENDIF
1575     RETURN
1576     END
1577 C=====
1578     SUBROUTINE ZONE(TIMEST)
1579 C=====
1580 C ZONE solves for the values of the variables (in array Y) describing a
1581 C building zone. The current values in Y are used as the starting point.
1582 C New values of Y at one time step in the future are calculated and
1583 C replace the old values of Y.
1584 C
1585 C Input variables:
1586 C TIMEST - The current value to be used for the major simulation time step.
1587 C
1588     INTEGER TIMEST
1589 C
1590 C Output variables: none
1591 C Procedures called: EULER,SURFAC,ZONE,ZONEFN,LOCAL
1592 C
1593     EXTERNAL ZONEFN
1594 C
1595 C Common blocks: STATE1,TIME
1596 C
1597     REAL          A
1598     LOGICAL       L
1599     COMMON / STATE1 / A(100),L(100)
1600 C
1601     INTEGER       DAY, HOUR, MINUTE, SECOND, ENDDAY, ENDHR, ENDMIN, ENDSEC
1602     COMMON / TIME / DAY, HOUR, MINUTE, SECOND, ENDDAY, ENDHR, ENDMIN, ENDSEC
1603 C
1604 C Local Variables:
1605 C
1606 C ADSTEP - Logical variable which if true, enables the use of the algorithm
1607 C to internally determine a minimum Euler timestep. If false, a
1608 C fixed Euler timestep is used.
1609 C DELMAX - The maximum change in state variable allowed for any of the
1610 C differential equations describing the zone within one Euler
1611 C timestep in degrees F. Theoretically, each differential equation
1612 C should have its own DELMAX. A simplification of this is used.
1613 C EISMIN - Minimum Euler timestep which can be used.
1614 C EISTEP - Current Euler timestep to be used in solving the differential
1615 C equations. determined from  $dt = dY_{max} / (dY/dt)$ , where  $dY_{max}$  is
1616 C DELMAX.
1617 C EUSTEP - Euler timestep to be used if ADSTEP is false.
1618 C FMAX - Magnitude of the largest  $dY/dt$  for all of the first order
1619 C differential equations used to describe
1620 C the zone.
1621 C ITIME - A time value which is a combination of the current minutes and
1622 C seconds of the time of day. Used to determine if time-dependent
1623 C functions should be recalculated.

```

```

1624 C NDEQNS - Number of variables in the zone model state vector which are
1625 C         defined by differential equations. These variables will be the
1626 C         first NDEQNS variables in the vector. The remaining variables
1627 C         will be defined by algebraic equations.
1628 C         in the state vector
1629 C NITER  - Number of iterations required to solve the zone differential
1630 C         equations for one major simulation timestep.
1631 C NSTVAR - Total number of variables in the zone model state vector.
1632 C STEP   - Floating point equivalent of the integer major timestep in the
1633 C         simulation.
1634 C Y      - Zone model state vector which is a subset of the simulation
1635 C         state vector. Elements are defined as:
1636 C         Y(1): zone dry bulb temperature (F)
1637 C         Y(2): zone humidity ratio
1638 C         Y(3): zone internal wall temperature at mass center (F)
1639 C         Y(4): zone interior wall temperature (F)
1640 C         Y(5): zone interior glass temperature (F)
1641 C
1642 LOGICAL ADSTEP
1643 PARAMETER (ADSTEP = .TRUE.,EUSTEP = 0.1)
1644 PARAMETER (NSTVAR = 3,NDEQNS = 3,DELMAX = 0.10, EISMIN = 0.001)
1645 INTEGER ITIME
1646 REAL Y(NSTVAR)
1647 EQUIVALENCE ( A(6),Y(1) )
1648 C
1649 C-----determine Euler time step-----
1650 C
1651 IF(ADSTEP)THEN
1652   FMAX = 0.0
1653   DO 900 I = 1,NDEQNS
1654     FMAX = AMAX1( ABS(ZONEFN(I,SECOND,NSTVAR,Y)) , FMAX )
1655     IF(FMAX.LE.1.E-8) FMAX = 1.E-8
1656 900 CONTINUE
1657   STEP = FLOAT(TIMEST)
1658   EISTEP = DELMAX / FMAX
1659   EISTEP = AMIN1( EISTEP, STEP )
1660   EISTEP = AMAX1( EISTEP, EISMIN )
1661   NITER = AINT( STEP / EISTEP )
1662   EISTEP = STEP / FLOAT(NITER)
1663 ELSE
1664   NITER = TIMEST / EUSTEP
1665   EISTEP = EUSTEP
1666 ENDIF
1667 C-----loop through internal iterations-----
1668 C
1669 ITIME = SECOND + MINUTE*60
1670 DO 1000 I = 1,NITER.
1671 C
1672 C-----solve NDEQNS simultaneous linear differential equations-----
1673 C
1674 CALL EULER(NDEQNS,ITIME,Y,EISTEP,ZONEFN,NSTVAR)
1675 C
1676 C-----solve algebraic equations for wall and glass surfaces-----
1677 C
1678 CALL SURFAC(NDEQNS,NSTVAR,Y,ITIME)
1679 C-----
1680 C Simulate the local zone space heating and cooling equipment. This is
1681 C an algebraic approximation.

```



```

1682 C
1683 CALL LOCAL(EISTEP)
1684 1000 CONTINUE
1685 C
1686 RETURN
1687 END
1688 C-----
1689 SUBROUTINE EULER(NDEQNS,TIME,Y,TIMEST,FNC,NSTVAR)
1690 C-----
1691 C This subroutine advances the system of NDEQNS ODE'S described by:
1692 C  $dY(I)/dt = FNC(I,t,Y(1),Y(2),\dots,Y(NDEQNS))$  by one step of size
1693 C TIMEST using the explicit Euler method. Y contains the dependent
1694 C variables and is overwritten with the new values.
1695 C
1696 C Input variables:
1697 C NDEQNS - Number of differential equations to be advanced.
1698 C TIME - A time value which must be distinct from the time value on
1699 C the previous call to EULER.
1700 C Y - contains the current values of the dependent variables.
1701 C TIMEST - The Euler timestep size.
1702 C FNC - The address of the function describing  $dY/dt$ .
1703 C NSTVAR - dimension of the Y array. Not all elements may be advanced.
1704 C
1705 REAL Y(NSTVAR),TIMEST
1706 INTEGER TIME
1707 C
1708 C Output variables: none
1709 C Y - contains the new value of the dependent variables.
1710 C
1711 C Procedures called: FNC
1712 C
1713 EXTERNAL FNC
1714 C
1715 C Common blocks: none
1716 C
1717 C-----advance dependent variables-----
1718 C
1719 DO 900 J = 1,NDEQNS
1720 Y(J) = Y(J) + TIMEST * FNC(J,TIME,NSTVAR,Y)
1721 900 CONTINUE
1722 C
1723 C-----the independent variable, time is advanced externally-----
1724 C
1725 RETURN
1726 END
1727 C-----
1728 SUBROUTINE SURFAC(NDEQNS,NSTVAR,Y,TIME)
1729 C-----
1730 C This subroutine is used to calculate the next value of the zone state
1731 C variables which can be calculated from an algebraic equation.
1732 C
1733 C Input variables:
1734 C NDEQNS - Number of variables in the zone model state vector which are
1735 C defined by differential equations. These variables will be the
1736 C first NDEQNS variables in the vector. The remaining variables
1737 C will be defined by algebraic equations.
1738 C in the state vector
1739 C NSTVAR - Total number of variables in the zone model state vector.

```

```

1740 C Y      - contains the current values of the dependent variables.
1741 C TIME   - A time value which is a combination of the current minutes and
1742 C        seconds of the time of day. Used to determine if time-dependent
1743 C        functions should be recalculated.
1744 C
1745     REAL Y(NSTVAR)
1746     INTEGER TIME
1747 C
1748 C Output variables:
1749 C Y      - contains the new value of the dependent variables.
1750 C        Y(4): zone interior wall temperature (F)
1751 C        Y(5): zone interior glass temperature (F)
1752 C
1753 C Procedures called: CNDCTI,CONVWA,GCNDCT,GCNVCT,GRADIA
1754 C Common blocks: STATE1
1755 C
1756     REAL          A
1757     LOGICAL       L
1758     COMMON / STATE1 / A(100),L(100)
1759 C
1760 C Local variables:
1761 C A41 - conduction component of heat transfer between interior of wall and
1762 C      zone air.
1763 C A42 - convection component of heat transfer between interior of wall and
1764 C      zone air.
1765 C A43 - radiation component of heat transfer between interior of wall and
1766 C      zone air.
1767 C A51 - conduction component of heat transfer between interior of wall and
1768 C      outside air.
1769 C A52 - convection component of heat transfer between interior of wall and
1770 C      outside air.
1771 C A53 - radiation component of heat transfer between interior of wall and
1772 C      outside air.
1773 C TI   - zone dry bulb temperature (F)
1774 C TOUTA - outside air dry bulb temperature (F)
1775 C TRN  - Mean radiant temperature in the zone (F)
1776 C TW   - zone internal wall temperature at mass center (F)
1777 C
1778     EQUIVALENCE ( A(1),TOUTA )
1779 C
1780     TI = Y(1)
1781     TW = Y(3)
1782     TRN = MRT(TIME)
1783 C-----wall interior surface temperature thermal energy balance ---
1784     A41 = CNDCTI(TIME)
1785     A42 = CONVWA(TIME)
1786     A43 = WRADIA(TIME)
1787     Y(4) = ( A41 * TW + A42 * TI + A43 * TRN + WSOLAR(TIME) ) /
1788     &      ( A41 + A42 + A43 )
1789 C-----glass interior surface temperature thermal energy balance ---
1790     A51 = GCNDCT(TIME)
1791     A52 = GCNVCT(TIME)
1792     A53 = GRADIA(TIME)
1793     IF(A51.NE.0.AND.A52.NE.0.AND.A53.NE.0)
1794     &Y(5) = ( A51 * TOUTA + A52 * TI + A53 * TRN ) /
1795     &      ( A51 + A52 + A53 )
1796 C
1797     RETURN

```

```

1798         END
1799 C-----
1800         FUNCTION ZONEFN(I,TIME,NSTVAR,Y)
1801 C-----
1802 C This function returns the right hand side of the linear ordinary
1803 C differential equation for the building zone model. This should be a
1804 C linear function of time, the set of dependent variables, boundary
1805 C conditions, and parameters.
1806 C
1807 C Input variables:
1808 C I      - Index of differential equation for which function is being
1809 C         determined.
1810 C TIME   - A time value which is a combination of the current minutes and
1811 C         seconds of the time of day. Used to determine if time-dependent
1812 C         functions should be recalculated
1813 C NSTVAR - Total number of variables in the zone model state vector.
1814 C Y      - Zone model state vector which is a subset of the simulation
1815 C         state vector.
1816 C
1817         INTEGER TIME
1818         REAL Y(NSTVAR)
1819 C
1820 C Output variables:
1821 C ZONEFN - right hand side of linear ODE.
1822 C
1823 C Procedures called: CNDCTI,CNDCTO,CONVGL,CONVWA,HUMRAT,INFILT,INFILW,
1824 C                   INTERN,MOIST,MASSA,MASSTH,MASSW,SOLAR,SUPLW,
1825 C                   SUPPLY
1826 C
1827         REAL CNDCTI,CNDCTO,CONVGL,CONVWA,HUMRAT,INFILT,INFILW,INTERN,
1828         &    MOIST,MASSA,MASSTH,MASSW,SOLAR,SUPLW,SUPPLY
1829 C
1830 C Common blocks: STATE1
1831 C
1832         REAL          A
1833         LOGICAL       L
1834         COMMON / STATE1 / A(100),L(100)
1835 C
1836 C Local Variables:
1837 C MAXW - Humidity ratio in the space at current temperature and 100% RH.
1838 C PRESS - Atmospheric pressure (assumed constant) in inches of Hg.
1839 C TI    - zone dry bulb temperature (F)
1840 C TIG   - zone interior glass temperature (F)
1841 C TIW   - zone interior wall temperature (F)
1842 C TOUTA - TEMPERATURE of outside air (F)
1843 C TW    - zone internal wall temperature at mass center (F)
1844 C WI    - zone humidity ratio
1845 C WOUTA - HUMIDITY RATIO of outside air (lbw/lbda)
1846 C
1847         PARAMETER (PRESS = 29.921)
1848         REAL MAXW
1849         EQUIVALENCE ( A(1),TOUTA ),( A(2),WOUTA )
1850 C
1851 C----- Zone Air Energy Balance -----
1852 C
1853         IF(I.EQ.1)THEN
1854             TI = Y(1)
1855             WI = Y(2)

```

```

1856         TIW = Y(4)
1857         TIG = Y(5)
1858         ZONEFN = (SUPPLY(TIME) +
1859         &         INFILT(TIME,TI,WI) * (TOUTA - TI) +
1860         &         CONVWA(TIME) * (TIW - TI) +
1861         &         CONVGL(TIME) * (TIG - TI) +
1862         &         INTERN(TIME))/ MASSSTH(TIME,WI,TI)
1863     C
1864     C----- Zone Air Moisture Balance -----
1865     C
1866         ELSE IF(1.EQ.2)THEN
1867             WI = Y(2)
1868             TI = Y(1)
1869             ZONEFN = (INFILW(TIME,TI) * (WOUTA - WI) +
1870             &         SUPPLW(TIME) +
1871             &         MOIST(TIME))/ MASSA(TIME,TI)
1872     C
1873     C If zone air reaches saturation, prevent further increase
1874     C
1875             MAXW = HUMRAT(TI,100.0,PRESS)
1876             IF(WI.GT.MAXW.AND.ZONEFN.GT.0.0) ZONEFN = 0.0
1877             IF(WI.LE.0.0.AND.ZONEFN.LT.0.0) ZONEFN = 0.0
1878     C
1879     C----- Wall Thermal Energy Balance -----
1880     C
1881         ELSE IF(1.EQ.3)THEN
1882             TW = Y(3)
1883             TIW = Y(4)
1884     C
1885     C NOTE: TOUTA in the equation below should be the solair temperature.....
1886     C
1887             ZONEFN = (CNDCTO(TIME) * (TOUTA - TW) +
1888             &         CNDCTI(TIME) * (TIW - TW) +
1889             &         SOLAR(TIME) )/ MASSW(TIME)
1890         ELSE
1891             ZONEFN = 0.
1892         ENDIF
1893     C
1894     C
1895         RETURN
1896         END
1897     C=====
1898         FUNCTION SUPPLY(TIME)
1899     C=====
1900     C Returns the amount of energy entering or being removed from the zone
1901     C by the HVAC system. This includes energy from the air handling unit
1902     C and the local conditioning equipment.
1903     C
1904     C UNITS: returns Btu/sec.
1905     C
1906     C Input variables:
1907     C TIME - Used to determine if time-dependent functions should be recalculated.
1908     C
1909         INTEGER TIME
1910     C
1911     C Output variables:
1912     C SUPPLY - Energy entering or being removed from the zone by the HVAC system.
1913     C

```

```

1914 REAL SUPPLY
1915 C
1916 C Procedures called: ENTHAL
1917 C Common blocks: STATE1
1918 C
1919 REAL A
1920 LOGICAL L
1921 COMMON / STATE1 / A(100),L(100)
1922 C
1923 C Local Variables:
1924 C FUNC - Holding variable for function return value.
1925 C HSA1 - enthalpy of the supply air in BTU/lbm
1926 C HSA2 - enthalpy of the hot deck (if existing) supply air Btu/lbm
1927 C HZA - enthalpy of zone air in Btu/lbm
1928 C MRA - mass flow rate of return air (lbm/s)
1929 C MSA1 - mass flow rate of supply air in Lbm/sec.
1930 C MSA2 - mass flow rate of hot deck (if existing) supply air in Lbm/sec.
1931 C NOINIT - logical variable, true if function has never been calculated.
1932 C OLD - time when function was last calculated.
1933 C PRESS - Atmospheric pressure (assumed constant) in inches of Hg.
1934 C QLOCH - energy added to the zone by local equipment during the time step.
1935 C QLOCC - energy taken from the zone by local equipment during the time step.
1936 C TIMEDP - logical parameter, true if function is time dependent.
1937 C TZA - temperature of zone air in degrees F.
1938 C WZA - humidity ratio of zone air in lbm water/ lbm dry air.
1939 C
1940 LOGICAL TIMEDP,NOINIT
1941 PARAMETER (TIMEDP = .TRUE.,PRESS = 29.921)
1942 INTEGER OLD
1943 DATA NOINIT/.TRUE./,OLD/-1/
1944 C
1945 REAL MRA,MSA1,MSA2
1946 EQUIVALENCE ( A(06),TZA ),( A(07),WZA ),( A(33),HSA1 )
1947 EQUIVALENCE ( A(51),MRA ),( A(62),QLOCH ),( A(63),QLOCC )
1948 EQUIVALENCE ( A(52),MSA1 )
1949 C
1950 IF(NOINIT.OR.(TIME.NE.OLD.AND.TIMEDP)) THEN
1951 C-----
1952 C THE FOLLOWING VALUES ARE FOR DUAL DECK (NOT IMPLEMENTED)
1953 MSA2 = 0.0
1954 HSA2 = 0.0
1955 C
1956 HZA = ENTHAL(TZA,WZA,PRESS)
1957 FUNC = (HSA1*MSA1) + (HSA2*MSA2) - (HZA*MRA) + QLOCH - QLOCC
1958 C-----
1959 OLD = TIME
1960 NOINIT = .FALSE.
1961 ENDIF
1962 SUPPLY = FUNC
1963 RETURN
1964 END
1965 C=====
1966 FUNCTION INFILT(TIME,TI,WI)
1967 C=====
1968 C Returns the infiltration coefficient for energy gain/loss due to
1969 C outside air infiltration.
1970 C
1971 C UNITS: returns Btu/sec F.

```

```

1972 C
1973 C Input variables:
1974 C TIME - Used to determine if time-dependent functions should be recalculated.
1975 C TI - Dry bulb temperature of zone air in F.
1976 C WI - Humidity ratio of zone air in lbm water/lbm dry air.
1977 C
1978 C INTEGER TIME
1979 C
1980 C Output variables:
1981 C INFILT - the infiltration coefficient for energy gain/loss.
1982 C
1983 C REAL INFILT
1984 C
1985 C Procedures called: AXRATE,CPAIR,RHQAIR
1986 C Common blocks: STATE1,SHELL
1987 C
1988 C REAL A
1989 C LOGICAL L
1990 C COMMON / STATE1 / A(100),L(100)
1991 C
1992 C EQUIVALENCE ( A(1),TOUTA ), ( A(2),WOUTA )
1993 C
1994 C REAL LENOW,LENIW,KOW,KIW
1995 C COMMON / SHELL / AREAW,HIWALL,HOWALL,LENOW,LENIW,KOW,KIW,CFUR,
1996 C & VOLAIR,CPGW,RHOGW,CPIW,RHOIW,AIREXC,RMGFUR
1997 C
1998 C Local Variables:
1999 C CPIA is specific heat of infiltration air (average of inside and outside)
2000 C FUNC - Holding variable for function return value.
2001 C NOINIT - logical variable, true if function has never been calculated.
2002 C OLD - time when function was last calculated.
2003 C RHOIA - density of infiltration air (average of inside and outside)
2004 C TIMEDP - logical parameter, true if function is time dependent.
2005 C TOUTA - the outdoor air temperature in degrees F.
2006 C WOUTA - the outdoor air humidity ratio in lbm water/lbm dry air.
2007 C
2008 C INTEGER OLD
2009 C LOGICAL TIMEDP,NOINIT
2010 C PARAMETER (TIMEDP = .TRUE.)
2011 C DATA NOINIT/.TRUE./,OLD/-1/
2012 C
2013 C IF(NOINIT.OR.(TIME.NE.OLD.AND.TIMEDP)) THEN
2014 C -----
2015 C
2016 C RHQAIR is a function returning the density of moist air
2017 C AXRATE is a function for the air exchange rate in air changes per second
2018 C VOLAIR is the volume of the zone air
2019 C
2020 C RHQAIR = (RHQAIR(WOUTA,TOUTA) + RHQAIR(WI,TI)) / 2.
2021 C CPIA = (CPAIR(WOUTA) + CPAIR(WI)) / 2.
2022 C FUNC = CPIA * RHOIA * VOLAIR * AXRATE(TIME,TI)
2023 C
2024 C -----
2025 C OLD = TIME
2026 C NOINIT = .FALSE.
2027 C ENDIF
2028 C INFILT = FUNC
2029 C RETURN

```

```

2030         END
2031 C-----
2032         FUNCTION AXRATE(TIME,TI)
2033 C-----
2034 C Returns the air exchange rate for a building zone.
2035 C
2036 C UNITS: returns air exchange rate in air changes per second
2037 C         TI and TO must be in degrees F
2038 C
2039 C Input variables:
2040 C TIME   - Used to determine if time-dependent functions should be recalculated.
2041 C TI     - Dry bulb temperature of zone air in F.
2042 C
2043         INTEGER TIME
2044 C
2045 C Output variables:
2046 C INFILT - the infiltration coefficient for energy gain/loss.
2047 C
2048         REAL AXRATE
2049 C
2050 C Procedures called: none
2051 C Common blocks: STATE1,SHELL
2052 C
2053         REAL          A
2054         LOGICAL       L
2055         COMMON / STATE1 / A(100),L(100)
2056 C
2057         EQUIVALENCE ( A(1),TOUTA ),( A(04),VWIND )
2058 C
2059         REAL          LENOW,LENIW,KOW,KIW
2060         COMMON / SHELL / AREAW,HIWALL,HOWALL,LENOW,LENIW,KOW,KIW,CFUR,
2061         &              VOLAIR,CPOW,RHOOW,CPIW,RHOIW,AIREXC,RMGFUR
2062 C
2063 C Local Variables:
2064 C FUNC   - Holding variable for function return value.
2065 C NOINIT - logical variable, true if function has never been calculated.
2066 C OLD    - time when function was last calculated.
2067 C TIMEDP - logical parameter, true if function is time dependent.
2068 C TOUTA  - the outdoor air temperature in degrees F.
2069 C VWIND  - the outdoor wind velocity in miles per hour.
2070 C
2071         INTEGER OLD
2072         LOGICAL TIMEDP,NOINIT
2073         PARAMETER (TIMEDP = .TRUE.)
2074         DATA NOINIT/.TRUE./,OLD/-1/
2075 C
2076         IF(NOINIT.OR.(TIME.NE.OLD.AND.TIMEDP)) THEN
2077 C-----
2078 C
2079 C AIREXC is the standard air exchange rate without the effects of wind
2080 C         and temperature differential in air changes per hour:
2081 C         = 1.5 for leaky building
2082 C         = 1.0 for standard building
2083 C         = 0.5 for modern type building
2084 C
2085         AXRATE=AIREXC/0.695*(0.15+0.013*VWIND+0.003*ABS(TOUTA-TI))
2086         FUNC = AXRATE / 3600.
2087 C-----

```

```

2088         OLD = TIME
2089         NOINIT = .FALSE.
2090     ENDIF
2091     AXRATE = FUNC
2092     RETURN
2093     END
2094 C=====
2095     FUNCTION CONVWA(TIME)
2096 C=====
2097 C Returns the convection coefficient for convection heat transfer between
2098 C the zone air and the wall (not including glass) surfaces enclosing the
2099 C zone.
2100 C
2101 C UNITS: returns Btu/s F.
2102 C
2103 C Input variables:
2104 C TIME - Used to determine if time-dependent functions should be recalculated.
2105 C
2106     INTEGER TIME
2107 C
2108 C Output variables:
2109 C CONVWA - the convection coefficient between zone air and wall.
2110 C
2111     REAL CONVWA
2112 C
2113 C Procedures called: none
2114 C Common blocks: SHELL
2115 C
2116     REAL                                LENOW, LENIW, KOW, KIW
2117     COMMON / SHELL / AREAW, HIWALL, HOWALL, LENOW, LENIW, KOW, KIW, CFUR,
2118     &                                VOLAIR, CPDW, RHOOW, CPIW, RHOIW, AIREXC, RMGFUR
2119 C
2120 C Local Variables:
2121 C FUNC - Holding variable for function return value.
2122 C NOINIT - logical variable, true if function has never been calculated.
2123 C OLD - time when function was last calculated.
2124 C TIMEDP - logical parameter, true if function is time dependent.
2125 C
2126     INTEGER OLD
2127     LOGICAL TIMEDP, NOINIT
2128     PARAMETER (TIMEDP = .FALSE.)
2129     DATA NOINIT/.TRUE./, OLD/-1/
2130 C
2131     IF(NOINIT.OR.(TIME.NE.OLD.AND.TIMEDP)) THEN
2132 C-----
2133 C HIWALL is the interior wall surface film coefficient
2134 C AREAW is the surface area of the wall exposed to the outside
2135 C
2136     FUNC = HIWALL * AREAW
2137 C-----
2138     OLD = TIME
2139     NOINIT = .FALSE.
2140     ENDIF
2141     CONVWA = FUNC
2142     RETURN
2143     END
2144 C=====
2145     FUNCTION CONVGL(TIME)

```



```

2146 C=====
2147 C Returns the convection coefficient for convection heat transfer between
2148 C the zone air and the glass (windows) surfaces enclosing the zone.
2149 C
2150 C UNITS: returns Btu/s F.
2151 C
2152 C     REAL CONVGL
2153 C
2154 C THIS FUNCTION NOT IMPLEMENTED (NO GLASS IN WALL).
2155 C
2156 C     CONVGL = 0.0
2157 C     RETURN
2158 C     END
2159 C=====
2160 C     FUNCTION INTERN(TIME)
2161 C=====
2162 C Returns the sensible internal gain in the zone from lighting, people and
2163 C equipment.
2164 C
2165 C UNITS: returns Btu/s.
2166 C
2167 C Input variables:
2168 C TIME - Used to determine if time-dependent functions should be recalculated.
2169 C
2170 C Output variables:
2171 C INTERN - sensible internal gain.
2172 C
2173 C     REAL INTERN
2174 C
2175 C Procedures called: SCHED
2176 C Common blocks: none
2177 C
2178 C     INTERN = SCHED(TIME, 'S')
2179 C     RETURN
2180 C     END
2181 C=====
2182 C     FUNCTION SCHED(TIME, SORL)
2183 C=====
2184 C This routine is used to determine the current values of sensible or
2185 C latent internal gains in the zone. The values of the gains are determined
2186 C from a schedule of occupancy , lighting and equipment usage.
2187 C
2188 C Input variables:
2189 C SORL - Character used to select the return of latent gain if 'L' and
2190 C       sensible gain if 'S'.
2191 C TIME - Used to determine if time-dependent functions should be recalculated.
2192 C
2193 C     INTEGER TIME
2194 C     CHARACTER*1 SORL
2195 C
2196 C Output variables: SCHED
2197 C SCHED - Contains either sensible internal gain in Btu/sec. or latent
2198 C         moisture gain in lbm water/sec.
2199 C
2200 C Procedures called: none
2201 C Common blocks: STATE1, TIME, USE
2202 C
2203 C     PARAMETER (NSCHED = 10)

```

```

2204         REAL          TSCHED,          SENSIB,          LATENT
2205         COMMON / USE / LSCHED,TSCHED(NSCHED),SENSIB(NSCHED),LATENT(NSCHED)
2206         &                ,PESENS(NSCHED),PELATE(NSCHED)
2207     C
2208         INTEGER          DAY,HOUR,MINUTE,SECOND,ENDDAY,ENDHR,ENDMIN,ENDSEC
2209         COMMON / TIME / DAY,HOUR,MINUTE,SECOND,ENDDAY,ENDHR,ENDMIN,ENDSEC
2210     C
2211         REAL          A
2212         LOGICAL       L
2213         COMMON / STATE1 / A(100),L(100)
2214     C
2215     C Local Variables:
2216     C
2217     C DHOOR - current hour of the day in decimal hours.
2218     C ISCHED - Index into schedule tables for sensible and latent loads for
2219     C          current time of day.
2220     C NOINIT - logical variable, true if function has never been calculated.
2221     C OCCUPY - STATUS zone occupied (.true.)/unoccupied (.false.).
2222     C OLD - time when function was last calculated.
2223     C TIMEDP - logical parameter, true if function is time dependent.
2224     C
2225         INTEGER OLD
2226         LOGICAL TIMEDP,NOINIT,OCCUPY
2227         PARAMETER (TIMEDP = .TRUE.)
2228         DATA NOINIT/.TRUE./,OLD/-1/
2229     C
2230         EQUIVALENCE ( L(11),OCCUPY )
2231     C
2232         IF(NOINIT.OR.(TIME.NE.OLD.AND.TIMEDP)) THEN
2233     C-----
2234     C Find row of schedule table for current time of day
2235     C
2236         DHOOR = HOUR + MINUTE/60. + SECOND/3600.
2237         DO 1000 I = 1,LSCHED
2238             IF(TSCHED(I).GT.DHOOR) GO TO 2000
2239     1000     CONTINUE
2240     2000     ISCHED = I - 1
2241             OLD = TIME
2242             NOINIT = .FALSE.
2243         ENDIF
2244     C
2245     C-----
2246     C Determine sensible or latent load from tables.
2247     C The following assumes that the sensible and moisture gains of people
2248     C are constant under all conditions. This is not strictly correct.
2249     C
2250         IF(SORL.EQ.'L')THEN
2251             SCHED = LATENT(ISCHED) + PELATE(ISCHED)
2252         ELSE IF(SORL.EQ.'S')THEN
2253             SCHED = SENSIB(ISCHED) + PESENS(ISCHED)
2254     C
2255     C-----
2256     C determine if building is occupied
2257     C
2258         IF(PESENS(ISCHED).GT.0)THEN
2259             OCCUPY = .TRUE.
2260         ELSE
2261             OCCUPY = .FALSE.

```

```

2262         ENDIF
2263     ELSE
2264         SCHED = 0.
2265     ENDIF
2266     RETURN
2267     END
2268 C-----
2269     FUNCTION MASSTH(TIME,WI,TI)
2270 C-----
2271 C Returns the thermal mass of the zone air and furnishings (not including
2272 C walls).
2273 C
2274 C UNITS: returns Btu/F.
2275 C
2276 C Input variables:
2277 C TIME - Used to determine if time-dependent functions should be recalculated.
2278 C TI - Dry bulb temperature of zone air in F.
2279 C WI - Humidity ratio of zone air in lbm water/lbm dry air.
2280 C
2281     INTEGER TIME
2282 C
2283 C Output variables:
2284 C MASSTH - the thermal mass of the zone air and furnishings in Btu/F.
2285 C
2286     REAL MASSTH
2287 C
2288 C Procedures called: CPAIR,RHOAIR
2289 C Common blocks: SHELL
2290 C
2291     REAL                                LENOW,LENIW,KOW,KIW
2292     COMMON / SHELL / AREAW,HIWALL,HOWALL,LENOW,LENIW,KOW,KIW,CFUR,
2293     & VOLAIR,CPGW,RHOGW,CPIW,RHOIW,AIREXC,RMGFUR
2294 C
2295 C Local Variables:
2296 C CAIR - thermal mass of the zone air.
2297 C FUNC - Holding variable for function return value.
2298 C NOINIT - logical variable, true if function has never been calculated.
2299 C OLD - time when function was last calculated.
2300 C TIMEDP - logical parameter, true if function is time dependent.
2301 C
2302     INTEGER OLD
2303     LOGICAL TIMEDP,NOINIT
2304     PARAMETER (TIMEDP = .TRUE.)
2305     DATA NOINIT/.TRUE./,OLD/-1/
2306 C
2307     IF(NOINIT.OR.(TIME.NE.OLD.AND.TIMEDP)) THEN
2308 C-----
2309 C VOLAIR is the volume of the zone air
2310 C RHOAIR is a function returning the density of moist air
2311 C CPAIR is a function returning the specific heat of moist air
2312 C CFUR is the thermal mass of the zone furnishings
2313 C
2314     CAIR = CPAIR(WI) * RHOAIR(WI, TI) * VOLAIR
2315     FUNC = CFUR + CAIR
2316 C-----
2317     OLD = TIME
2318     NOINIT = .FALSE.
2319     ENDIF

```

```

2320     MASSTH = FUNC
2321     RETURN
2322     END
2323 C=====
2324     FUNCTION INFILW(TIME,TI,WI)
2325 C=====
2326 C Returns the infiltration coefficient for moisture gain/loss by outside
2327 C air infiltration.
2328 C
2329 C UNITS: returns lbm dry air/second.
2330 C
2331 C Input variables:
2332 C TIME - Used to determine if time-dependent functions should be recalculated.
2333 C TI - Dry bulb temperature of zone air in F.
2334 C WI - Humidity ratio of zone air in lbm water/lbm dry air.
2335 C
2336     INTEGER TIME
2337 C
2338 C Output variables:
2339 C INFILW - the mass flow rate of dry air from outside to inside which
2340 C carries water vapor with it. The actual change in moisture will
2341 C be assumed to be: INFILW times the indoor-outdoor humidity ratio
2342 C differential.
2343 C
2344     REAL INFILW
2345 C
2346 C Procedures called: AXRATE,RHOAIR
2347 C Common blocks: SHELL,STATE1
2348 C
2349     REAL A
2350     LOGICAL L
2351     COMMON / STATE1 / A(100),L(100)
2352 C
2353     REAL LENOW,LENIW,KOW,KIW
2354     COMMON / SHELL / AREAW,HIWALL,HOWALL,LENOW,LENIW,KOW,KIW,CFUR,
2355     & VOLAIR,CPOW,RHOOW,CPIW,RHOIW,AIREXC,RMGFUR
2356 C
2357 C Local Variables:
2358 C FUNC - Holding variable for function return value.
2359 C NOINIT - logical variable, true if function has never been calculated.
2360 C OLD - time when function was last calculated.
2361 C RHOIA - density of DRY infiltration air (average of inside and outside)
2362 C TIMEDP - logical parameter, true if function is time dependent.
2363 C TOUTA - the outdoor air temperature in degrees F.
2364 C
2365     INTEGER OLD
2366     LOGICAL TIMEDP,NOINIT
2367     PARAMETER (TIMEDP = .TRUE.)
2368     EQUIVALENCE ( A(1),TOUTA )
2369     DATA NOINIT/.TRUE./,OLD/-1/
2370 C
2371     IF(NOINIT.OR.(TIME.NE.OLD.AND.TIMEDP)) THEN
2372 C-----
2373 C
2374 C RHOAIR is a function returning the density of moist air
2375 C AXRATE is a function for the air exchange rate in air changes per second
2376 C VOLAIR is the volume of the zone air.
2377 C

```

```

2378      RHOA = (RHOAIR(0.0,TOUTA) + RHOAIR(0.0,TI)) / 2.
2379      FUNC = RHOA * VOLAIR * AXRATE(TIME,TI)
2380 C-----
2381      OLD = TIME
2382      NOINIT = .FALSE.
2383      ENDIF
2384      INFILW = FUNC
2385      RETURN
2386      END
2387 C=====
2388      FUNCTION SUPPLW(TIME)
2389 C=====
2390 C Returns the net amount of water entering the zone from the air handling
2391 C unit. If negative, water is being extracted from the zone.
2392 C
2393 C UNITS: lbm water/ second.
2394 C
2395 C Input variables:
2396 C TIME - Used to determine if time-dependent functions should be recalculated.
2397 C
2398      INTEGER TIME
2399 C
2400 C Output variables:
2401 C SUPPLW - water entering or being removed from the zone by the HVAC system.
2402 C
2403      REAL SUPPLW
2404 C
2405 C Procedures called: none
2406 C Common blocks: STATE1
2407 C
2408      REAL          A
2409      LOGICAL       L
2410      COMMON / STATE1 / A(100),L(100)
2411 C
2412 C Local Variables:
2413 C FUNC - Holding variable for function return value.
2414 C MRA - mass flow rate of return air (lbm/s)
2415 C MSA1 - mass flow rate of supply air in LBm/sec.
2416 C MSA2 - mass flow rate of hot deck (if existing) supply air in LBm/sec.
2417 C NOINIT - logical variable, true if function has never been calculated.
2418 C OLD - time when function was last calculated.
2419 C PRESS - Atmospheric pressure (assumed constant) in inches of Hg.
2420 C TIMEDP - logical parameter, true if function is time dependent.
2421 C WRA - humidity ratio of return air in lbm water/ lbm dry air.
2422 C WSA1 - humidity ratio of supply air in lbm water/ lbm dry air.
2423 C
2424      INTEGER OLD
2425      LOGICAL TIMEDP,NOINIT
2426      PARAMETER (TIMEDP = .TRUE.)
2427      DATA NOINIT/.TRUE./,OLD/-1/
2428 C
2429      REAL MRA,MSA1,MSA2,WRA,WSA1,WSA2
2430      EQUIVALENCE ( A(22),WRA ),( A(51),MRA )
2431      EQUIVALENCE ( A(34),WSA1 ),( A(52),MSA1 )
2432 C
2433 C
2434      IF(NOINIT.OR.(TIME.NE.OLD.AND.TIMEDP)) THEN
2435 C-----

```

```

2436 C THE FOLLOWING VALUES ARE FOR DUAL DECK (NOT IMPLEMENTED)
2437 MSA2 = 0.0
2438 WSA2 = 0.0
2439 C
2440 C NOTE: this is approximate, since MRA and MSA1 are in lb/sec of moist air.
2441 C
2442 FUNC = (WSA1 * MSA1) + (WSA2 * MSA2) - (WRA * MRA)
2443 C-----
2444 OLD = TIME
2445 NOINIT = .FALSE.
2446 ENDIF
2447 SUPPLW = FUNC
2448 RETURN
2449 END
2450 C=====
2451 FUNCTION MOIST(TIME)
2452 C=====
2453 C Returns the amount of water generated in the zone by people and equipment.
2454 C
2455 C UNITS: lbm water/ second.
2456 C
2457 C Input variables:
2458 C TIME - Used to determine if time-dependent functions should be recalculated.
2459 C
2460 C Output variables:
2461 C MOIST - moisture gain from people and equipment.
2462 C
2463 REAL MOIST
2464 C
2465 C Procedures called: SCHED
2466 C Common blocks: none
2467 C
2468 MOIST = SCHED(TIME, 'L')
2469 RETURN
2470 END
2471 C=====
2472 FUNCTION MASSA(TIME, TI)
2473 C=====
2474 C Returns the mass of the zone air which can absorb and release water
2475 C vapor. (mass of dry air)
2476 C
2477 C UNITS: returns lbm of dry air.
2478 C
2479 C Input variables:
2480 C TIME - Used to determine if time-dependent functions should be recalculated.
2481 C TI - Dry bulb temperature of zone air in F.
2482 C
2483 INTEGER TIME
2484 C
2485 C Output variables:
2486 C MASSA - the infiltration coefficient for energy gain/loss.
2487 C
2488 REAL MASSA
2489 C
2490 C Procedures called: RHOAIR
2491 C Common blocks: SHELL
2492 C
2493 REAL LENO, LENIW, KOW, KIW

```

```

2494 COMMON / SHELL / AREAW, HIWALL, HOWALL, LENOW, LENIW, KOW, KIW, CFUR,
2495 & VOLAIR, CPOW, RHODW, CPIW, RHOIW, AIREXC, RMGFUR
2496
2497 C Local Variables:
2498 C FUNC - Holding variable for function return value.
2499 C NOINIT - logical variable, true if function has never been calculated.
2500 C OLD - time when function was last calculated.
2501 C TIMEDP - logical parameter, true if function is time dependent.
2502 C WI - zone humidity ratio, set to zero for dry air.
2503 C
2504 C INTEGER OLD
2505 C LOGICAL TIMEDP, NOINIT
2506 C PARAMETER (TIMEDP = .TRUE.)
2507 C DATA NOINIT/.TRUE./, OLD/-1/
2508 C
2509 C IF (NOINIT .OR. (TIME .NE. OLD .AND. TIMEDP)) THEN
2510 C-----
2511 C VOLAIR is the volume of the zone air.
2512 C RHOAIR is a function returning the density of moist air.
2513 C RMGFUR is a factor used to produce the effective air mass for moisture
2514 C absorption by the zone furnishings and air.
2515 C
2516 C WI = 0.0
2517 C FUNC = RHOAIR(WI, TI) * VOLAIR * RMGFUR
2518 C-----
2519 C OLD = TIME
2520 C NOINIT = .FALSE.
2521 C ENDIF
2522 C MASSA = FUNC
2523 C RETURN
2524 C END
2525 C=====
2526 C FUNCTION CNDCTG(TIME)
2527 C=====
2528 C Returns the coefficient for thermal conductance between the mass center
2529 C of the wall and the environment outside of the building.
2530 C
2531 C UNITS: returns Btu/second F.
2532 C
2533 C Input variables:
2534 C TIME - Used to determine if time-dependent functions should be recalculated.
2535 C
2536 C INTEGER TIME
2537 C
2538 C Output variables:
2539 C CNDCTG - conductance coefficient between outside air and wall mass center.
2540 C
2541 C REAL CNDCTG
2542 C
2543 C Procedures called: none
2544 C Common blocks: SHELL
2545 C
2546 C REAL LENOW, LENIW, KOW, KIW
2547 C COMMON / SHELL / AREAW, HIWALL, HOWALL, LENOW, LENIW, KOW, KIW, CFUR,
2548 C & VOLAIR, CPOW, RHODW, CPIW, RHOIW, AIREXC, RMGFUR
2549 C
2550 C Local Variables:
2551 C FUNC - Holding variable for function return value.

```

```

2552 C NOINIT - logical variable, true if function has never been calculated.
2553 C OLD - time when function was last calculated.
2554 C TIMEDP - logical parameter, true if function is time dependent.
2555 C
2556 C INTEGER OLD
2557 C LOGICAL TIMEDP,NOINIT
2558 C PARAMETER (TIMEDP = .FALSE.)
2559 C DATA NOINIT/.TRUE./,OLD/-1/
2560 C
2561 C IF(NOINIT.OR.(TIME.NE.OLD.AND.TIMEDP)) THEN
2562 C-----
2563 C HOWALL is the surface film coefficient for the outside wall
2564 C LENOW is the thickness of the wall outside of the mass center
2565 C KOW is the thermal conductivity of the wall outside of the mass
2566 C center
2567 C AREAW is the area of the wall exposed to the outside
2568 C
2569 C FUNC = 1 / (1/HOWALL + LENOW/KOW) * AREAW
2570 C-----
2571 C OLD = TIME
2572 C NOINIT = .FALSE.
2573 C ENDIF
2574 C CNDCT0 = FUNC
2575 C RETURN
2576 C END
2577 C=====
2578 C FUNCTION CNDCTI(TIME)
2579 C=====
2580 C Returns the coefficient for thermal conductance between the mass center
2581 C of the wall and the inner wall surface surrounding the zone.
2582 C
2583 C UNITS: returns Btu/second F.
2584 C
2585 C Input variables:
2586 C TIME - Used to determine if time-dependent functions should be recalculated.
2587 C
2588 C INTEGER TIME
2589 C
2590 C Output variables:
2591 C CNDCTI - conductance coefficient between zone air and wall mass center.
2592 C
2593 C REAL CNDCTI
2594 C
2595 C Procedures called: none
2596 C Common blocks: SHELL
2597 C
2598 C REAL LENOW, LENIW, KOW, KIW
2599 C COMMON / SHELL / AREAW, HIWALL, HOWALL, LENOW, LENIW, KOW, KIW, CFUR,
2600 C & VOLAIR, CPOW, RHOGW, CPIW, RHOIW, AIREXC, RMGFUR
2601 C
2602 C Local Variables:
2603 C FUNC - Holding variable for function return value.
2604 C NOINIT - logical variable, true if function has never been calculated.
2605 C OLD - time when function was last calculated.
2606 C TIMEDP - logical parameter, true if function is time dependent.
2607 C
2608 C INTEGER OLD
2609 C LOGICAL TIMEDP,NOINIT

```



```

2610     PARAMETER (TIMEDP = .FALSE.)
2611     DATA NOINIT/.TRUE./,OLD/-1/
2612
2613     C
2614     IF(NOINIT.OR.(TIME.NE.OLD.AND.TIMEDP)) THEN
2615     C-----
2615     C     KIW is the thermal conductivity of the wall inside of the mass
2616     C     center
2617     C     LENIW is the thickness of the wall inside the mass center
2618     C     AREAW is the surface area of wall exposed to the outside
2619     C
2620     C     FUNC = KIW / LENIW * AREAW
2621     C-----
2622     C     OLD = TIME
2623     C     NOINIT = .FALSE.
2624     C     ENDIF
2625     C     CNDCTI = FUNC
2626     C     RETURN
2627     C     END
2628     C-----
2629     C     FUNCTION SOLAR(TIME)
2630     C-----
2631     C Returns the amount of solar heat gain through the windows in the zone.
2632     C
2633     C UNITS: returns Btu/second.
2634     C
2635     C     REAL SOLAR
2636     C
2637     C
2638     C THIS FUNCTION NOT IMPLEMENTED (NO GLASS IN WALL).
2639     C
2640     C     SOLAR = 0.0
2641     C     RETURN
2642     C     END
2643     C-----
2644     C     FUNCTION MASSW(TIME)
2645     C-----
2646     C Returns the thermal capacitance of the wall surrounding the zone.
2647     C
2648     C UNITS: returns Btu/F
2649     C
2650     C Input variables:
2651     C TIME - Used to determine if time-dependent functions should be recalculated.
2652     C
2653     C     INTEGER TIME
2654     C
2655     C Output variables:
2656     C MASSW - thermal capacitance of wall.
2657     C
2658     C     REAL MASSW
2659     C
2660     C Procedures called: none
2661     C Common blocks: SHELL
2662     C
2663     C     REAL
2664     C     COMMON / SHELL / AREAW,HIWALL,HOWALL,LENOW,LENIW,KOW,KIW,CFUR,
2665     C     & VOLAIR,CPOW,RHOOW,CPIW,RHOIW,AIREXC,RMGFUR
2666     C
2667     C Local Variables:

```

```

2668 C FUNC - Holding variable for function return value.
2669 C NOINIT - logical variable, true if function has never been calculated.
2670 C OLD - time when function was last calculated.
2671 C TIMEDP - logical parameter, true if function is time dependent.
2672 C
2673     INTEGER OLD
2674     LOGICAL TIMEDP,NOINIT
2675     PARAMETER (TIMEDP = .FALSE.)
2676     DATA NOINIT/.TRUE./,OLD/-1/
2677 C
2678     IF(NOINIT.OR.(TIME.NE.OLD.AND.TIMEDP)) THEN
2679 C-----
2680 C CPOW is the specific heat of the wall outside of mass center
2681 C RHOOW is the density of the wall outside of mass center.
2682 C LENOW is the thickness of the wall outside of mass center.
2683 C CPIW is the specific heat of the wall inside of mass center
2684 C RHOIW is the density of the wall inside of mass center.
2685 C LENIW is the thickness of the wall inside of mass center.
2686 C AREAW is the surface area of wall exposed to the outside
2687 C
2688     FUNC = (CPOW * RHOOW * LENOW + CPIW * RHOIW * LENIW) * AREAW
2689 C-----
2690     OLD = TIME
2691     NOINIT = .FALSE.
2692     ENDIF
2693     MASSW = FUNC
2694     RETURN
2695     END
2696 C=====
2697     FUNCTION WRADIA(TIME)
2698 C=====
2699 C Returns the coefficient for radiation heat transfer between the wall and
2700 C any other walls or furnishings (at room mean radiant temperature).
2701 C
2702     REAL WRADIA
2703 C
2704 C THIS FUNCTION NOT IMPLEMENTED (no radiation assumed).
2705 C
2706     WRADIA = 0.0
2707     RETURN
2708     END
2709 C=====
2710     FUNCTION WSOLAR(TIME)
2711 C=====
2712 C Returns the amount of solar energy entering the zone which is absorbed
2713 C by the wall surface.
2714 C
2715     REAL WSOLAR
2716 C
2717 C THIS FUNCTION NOT IMPLEMENTED (no windows assumed).
2718 C
2719     WSOLAR = 0.0
2720     RETURN
2721     END
2722 C=====
2723     FUNCTION GCNDCT(TIME)
2724 C=====
2725 C Returns the coefficient for conductance through the glass in the zone

```

```

2726 C from the inner wall surface and the environment outside the building.
2727 C
2728 REAL GCNDCT
2729 C
2730 C
2731 C THIS FUNCTION NOT IMPLEMENTED (no windows assumed).
2732 C
2733 GCNDCT = 0.0
2734 RETURN
2735 END
2736 C=====
2737 FUNCTION GCNVCT(TIME)
2738 C=====
2739 C Returns the coefficient for convection between the inner glass surface
2740 C and the air in the zone.
2741 C
2742 REAL GCNVCT
2743 C
2744 C
2745 C THIS FUNCTION NOT IMPLEMENTED (no windows assumed).
2746 C
2747 GCNVCT = 0.0
2748 RETURN
2749 END
2750 C=====
2751 FUNCTION GRADIA(TIME)
2752 C=====
2753 C Returns the coefficient for radiation heat transfer between the glass
2754 C and any other surfaces or furnishings in the zone.
2755 C
2756 REAL GRADIA
2757 C
2758 C
2759 C THIS FUNCTION NOT IMPLEMENTED (no windows assumed).
2760 C
2761 GRADIA = 0.0
2762 RETURN
2763 END
2764 C=====
2765 FUNCTION MRT(TIME)
2766 C=====
2767 C Returns the value of the mean radiant temperature in the zone.
2768 C
2769 REAL MRT
2770 C
2771 C
2772 C THIS FUNCTION NOT IMPLEMENTED (no radiation transfer assumed).
2773 C
2774 MRT = 0.0
2775 RETURN
2776 END
2777 C=====
2778 SUBROUTINE COMFRT(TIMEST)
2779 C=====
2780 C This routine is used to determine if a zone space is within the limits
2781 C of comfort. Comfort is defined in accordance with ASHRAE standard
2782 C 55-74 as being acceptable when the dry bulb temperature and humidity
2783 C ratio are within certain limits. The check of comfort is only performed

```

```

2784 C If the building zone is occupied, as determined by the logical variable
2785 C OCCUPY in the state vector.
2786 C
2787 C Input variables:
2788 C TIMEST - The current value to be used for the major simulation time step.
2789 C
2790     INTEGER TIMEST
2791 C
2792 C Output variables: none
2793 C Procedures called: RELHUM
2794 C Common blocks: STATE1
2795 C
2796     REAL          A
2797     LOGICAL       L
2798     COMMON / STATE1 / A(100),L(100)
2799 C
2800 C Local Variables:
2801 C OCCUPY - STATUS zone occupied (.true.)/unoccupied (.false.).
2802 C OUT - Logical variable which is true when comfort is outside of limits.
2803 C PRESS - Atmospheric pressure (assumed constant) in inches of Hg.
2804 C RHAMAX - COMFORT relative humidity at maximum temperature (occupied) (%)
2805 C RHAMIN - COMFORT relative humidity at minimum temperature (occupied) (%)
2806 C SECOUT - COMFORT seconds outside of comfort range
2807 C TCLEFT - Minimum temperature comfort limit, function of humidity ratio.
2808 C TCRITE - Maximum temperature comfort limit, function of humidity ratio.
2809 C TZA - temperature of zone air in degrees F.
2810 C TZMAX - COMFORT maximum dry bulb temperature during occupied period (F).
2811 C TZMIN - COMFORT minimum dry bulb temperature during occupied period (F).
2812 C WZA - humidity ratio of zone air in lbm water/ lbm dry air.
2813 C
2814     PARAMETER (PRESS = 29.921)
2815     LOGICAL OCCUPY,OUT
2816     EQUIVALENCE ( L(11),OCCUPY ), ( A(06),TZA ), ( A(07),WZA )
2817     EQUIVALENCE ( A(90),SECOUT ), ( A(91),TZMAX ), ( A(92),TZMIN )
2818     EQUIVALENCE ( A(93),RHAMAX ), ( A(94),RHAMIN )
2819 C
2820 C-----
2821 C Test for conditions outside of comfort limits
2822 C
2823     IF(OCCUPY) THEN
2824         TCLEFT = 72.58 - 131.58*(WZA)
2825         TCRITE = 81.73 - 394.74*(WZA)
2826         IF(WZA.GT.0.012.OR.WZA.LT.0.0044) THEN
2827             OUT = .TRUE.
2828         ELSE IF(TZA.LT.TCLEFT.OR.TZA.GT.TCRITE) THEN
2829             OUT = .TRUE.
2830         ELSE
2831             OUT = .FALSE.
2832         ENDIF
2833         IF(OUT) SECOUT = SECOUT + TIMEST
2834 C
2835 C-----
2836 C Test for max and min space conditions
2837 C
2838     IF(TZA.GT.TZMAX) THEN
2839         TZMAX = TZA
2840         RHAMAX = RELHUM(TZA,WZA,PRESS)
2841     ELSE IF(TZA.LT.TZMIN) THEN

```

```

2842         TZMIN = TZA
2843         RHAMIN = RELHUM(TZA,WZA,PRESS)
2844     ENDIF
2845 ENDIF
2846 C
2847 C-----
2848 C Test for max rate of temperature change
2849 C
2850 C NOT IMPLEMENTED
2851 C
2852     RETURN
2853     END
2854 C=====
2855     SUBROUTINE CMPILE(TIMEST)
2856 C=====
2857 C This routine is used to compile cumulative totals of energy use.
2858 C Four types of energy use are compiled. These are:
2859 C 1. Fan Energy Use
2860 C 2. Economizer Energy = energy provided by economizer.
2861 C 3. Load Energy = energy required to heat or cool air to space entering
2862 C         condition.
2863 C 4. Reheat/recool Energy = energy used to reheat and recool air for space
2864 C         temperature control.
2865 C Total Thermal Energy Use is also compiled. All thermal energy values have
2866 C a heating component and a cooling component.
2867 C
2868 C Input variables:
2869 C TIMEST - The current value to be used for the major simulation time step.
2870 C
2871     INTEGER TIMEST
2872 C
2873 C Output variables: none
2874 C Procedures called: RHGAIR,SHOTS
2875 C Common blocks: HVAC,STATE1
2876
2877     LOGICAL          DUDECK,HUMCON,RAFAN,VAV
2878     COMMON / HVAC / DUDECK,HUMCON,RAFAN,VAV,VOLSAF,CAPLOC,PGAINL,
2879     &                TCTHSE,QRAFAN,FVAMIN,QSAFAN,HCAPI,CCAPI,DTSETB,
2880     &                POWSAF,POWRAF,GAINIL
2881 C
2882     REAL             A          L
2883     LOGICAL          L
2884     COMMON / STATE1 / A(100),L(100)
2885 C
2886 C Local Variables:
2887 C EACTC - actual energy for cooling (Btu)
2888 C EACTH - actual energy for heating (Btu)
2889 C EECOC - COOLING ENERGY (economizer) (Btu)
2890 C EECOH - HEATING ENERGY (economizer) (Btu)
2891 C EFANS - ELECTRICAL ENERGY to fans (kWh)
2892 C ELOADC - COOLING ENERGY (load) (Btu)
2893 C ELOADH - HEATING ENERGY (load) (Btu)
2894 C ELOCC - COOLING ENERGY from local zone equipment (Btu)
2895 C ELOCH - HEATING ENERGY to local zone equipment (Btu)
2896 C EREC - COOLING ENERGY (recool) (Btu)
2897 C EREH - HEATING ENERGY (reheat) (Btu)
2898 C EREQC - COOLING ENERGY (requirements) (Btu)
2899 C EREQH - HEATING ENERGY (requirements) (Btu)

```

```

2900 C HMA - ENTHALPY of mixed air (BTU/lb)
2901 C HMC - ENTHALPY of return air plus ventilation air(BTU/lb)
2902 C HSA1 - ENTHALPY of supply air (BTU/lb) cold deck
2903 C HSA2 - ENTHALPY of hot deck (if existing) supply air in (Btu/lbm)
2904 C MRA - mass flow rate of return air (lbm/s)
2905 C MSA1 - mass flow rate of supply air in LBm/sec.
2906 C MSA2 - mass flow rate of hot deck (if existing) supply air in LBm/sec.
2907 C POWR - electric power to return air fan (kW)
2908 C POWS - electric power to supply air fan (kW)
2909 C QACT1 - actual energy for cold deck (or single deck) (Btu).
2910 C QACT2 - actual energy for hot deck (if existing) (Btu).
2911 C QLOAD - load energy (Btu), may be negative for energy extraction (cooling)
2912 C QREQ1 - requirements energy for cold deck (or single deck) (Btu).
2913 C QREQ2 - requirements energy for hot deck (if existing) (Btu).
2914 C RVOLR - ratio of current return volume flow rate to rated volume flow rate
2915 C RVOLS - ratio of current supply volume flow rate to rated volume flow rate
2916 C TRA - TEMPERATURE of return air (F)
2917 C TSA1 - TEMPERATURE of supply air (F) cold deck
2918 C TSA2 - Temperature of hot deck (if existing) supply air (F)
2919 C VRA - Volume flow rate of return air in ft3/sec.
2920 C VSA - Volume flow rate of supply air in ft3/sec.
2921 C WRA - HUMIDITY RATIO of return air (lbw/lbda)
2922 C WSA1 - HUMIDITY RATIO of supply air (lbw/lbda) cold deck
2923 C WSA2 - Humidity ratio of hot deck (if existing) supply air (lbw/lbda)
2924 C
2925 REAL MSA1,MSA2,MRA
2926 EQUIVALENCE ( A(52),MSA1 ),( A(33),HSA1 ),( A(24),HMA )
2927 EQUIVALENCE ( A(64),EREQH ),( A(65),EREQC ),( A(35),HMC )
2928 EQUIVALENCE ( A(66),EECOH ),( A(67),EECC ),( A(68),ELGADH )
2929 EQUIVALENCE ( A(69),ELGADC ),( A(73),ELGCH ),( A(70),EREH )
2930 EQUIVALENCE ( A(71),EREC ),( A(72),EFANS ),( A(34),WSA1 )
2931 EQUIVALENCE ( A(32),TSA1 ),( A(20),TRA ),( A(22),WRA )
2932 EQUIVALENCE ( A(51),MRA ),( A(74),ELOCC )
2933 C
2934 C-----
2935 C Fan Energy
2936 C
2937 C These calculations are based on fan law number 3 from ASHRAE equipment
2938 C handbook 1983, page 3.5, table 2.
2939 C
2940 MSA2 = 0.
2941 WSA2 = 0.
2942 TSA2 = 0.
2943 IF(MSA1.GT.0.0) THEN
2944 VSA = MSA1 / RHOAIR(WSA1,TSA1)
2945 IF(MSA2.GT.0.0) VSA = VSA + MSA2 / RHOAIR(WSA2,TSA2)
2946 RVOLS = VSA / VOLSAF
2947 POWS = POWSAF * RVOLS * RVOLS * RVOLS
2948 EFANS = EFANS + POWS * TIMEST / 3600.
2949 IF(RAFAN) THEN
2950 VRA = MRA / RHOAIR(WRA,TRA)
2951 RVOLR = VRA / VOLSAF
2952 POWR = POWRAF * RVOLR * RVOLR * RVOLR
2953 EFANS = EFANS + POWR * TIMEST / 3600.
2954 ENDIF
2955 ENDIF
2956 C
2957 C-----

```

```

2958 C Requirements Energy
2959 C
2960     MSA2 = 0.0
2961     HSA2 = 0.0
2962 C
2963     QREQ1 = MSA1 * (HSA1 - HMC) * TIMEST
2964     QREQ2 = MSA2 * (HSA2 - HMC) * TIMEST
2965     IF(QREQ1.GT.0.) THEN
2966         EREQH = EREQH + QREQ1
2967     ELSE
2968         EREQC = EREQC + QREQ1
2969     ENDIF
2970     IF(QREQ2.GT.0.) THEN
2971         EREQH = EREQH + QREQ2
2972     ELSE
2973         EREQC = EREQC + QREQ2
2974     ENDIF
2975
2976 C-----
2977 C Actual Energy
2978 C
2979     QACT1 = MSA1 * (HSA1 - HMA) * TIMEST
2980     QACT2 = MSA2 * (HSA2 - HMA) * TIMEST
2981     IF(QACT1.GT.0.) THEN
2982         EACTH = EACTH + QACT1
2983     ELSE
2984         EACTC = EACTC + QACT1
2985     ENDIF
2986     IF(QACT2.GT.0.) THEN
2987         EACTH = EACTH + QACT2
2988     ELSE
2989         EACTC = EACTC + QACT2
2990     ENDIF
2991
2992 C-----
2993 C Economizer Energy
2994 C
2995     EECOH = EREQH - EACTH
2996     EEOCC = EREQC - EACTC
2997
2998 C
2999 C-----
3000 C Load Energy
3001 C
3002     QLOAD = ( (MSA1*HSA1 + MSA2*HSA2) - (MSA1+MSA2)*HMA ) * TIMEST
3003     IF(QLOAD.GT.0.) THEN
3004         ELOADH = ELOADH + QLOAD
3005     ELSE
3006         ELOADC = ELOADC + QLOAD
3007     ENDIF
3008
3009 C-----
3010 C Reheat/Recool Energy
3011 C
3012     EREH = EACTH - ELOADH + ELOCH
3013     EREC = EACTC - ELOADC + ELOCC
3014
3015 C*****

```

```

3016          CALL SHOTS
3017 C*****
3018          RETURN
3019          END
3020 C*****
3021          SUBROUTINE REPORT
3022 C*****
3023 C This routine is used to output the final energy usage numbers at the
3024 C conclusion of a test.
3025 C The output is current sent to the console but should eventually be
3026 C transmitted to a file for further analysis by an additional program.
3027 C
3028 C Input variables: none
3029 C Output variables: none
3030 C Procedures called: none
3031 C Common blocks: PFILES, STATE1, UNITS
3032 C
3033          INTEGER          CLU, FLU, ILU, PLU
3034          COMMON / UNITS / CLU, FLU, ILU, PLU
3035 C
3036          CHARACTER*15     HVFILE, USFILE, CLFILE, SHFILE, INFILE, SEASON
3037          COMMON / PFILES / HVFILE, USFILE, CLFILE, SHFILE, INFILE, SEASON
3038 C
3039          REAL              A
3040          LOGICAL           L
3041          COMMON / STATE1 / A(100), L(100)
3042 C
3043 C Local Variables:
3044 C
3045 C EECOC - COOLING ENERGY (economizer) (Btu)
3046 C EECOH - HEATING ENERGY (economizer) (Btu)
3047 C EFANS - ELECTRICAL ENERGY to fans (kWh)
3048 C ELOADC - COOLING ENERGY (load) (Btu)
3049 C ELOADH - HEATING ENERGY (load) (Btu)
3050 C EREC - COOLING ENERGY (recool) (Btu)
3051 C EREH - HEATING ENERGY (reheat) (Btu)
3052 C EREQC - COOLING ENERGY (requirements) (Btu)
3053 C EREQH - HEATING ENERGY (requirements) (Btu)
3054 C RHAMAX - COMFORT relative humidity at maximum temperature (occupied) (%)
3055 C RHAMIN - COMFORT relative humidity at minimum temperature (occupied) (%)
3056 C SECOU - COMFORT seconds outside of comfort range
3057 C TZMAX - COMFORT maximum dry bulb temperature during occupied period (F).
3058 C TZMIN - COMFORT minimum dry bulb temperature during occupied period (F).
3059 C
3060          EQUIVALENCE ( A(64), EREQH ), ( A(65), EREQC ), ( A(72), EFANS )
3061          EQUIVALENCE ( A(66), EECOH ), ( A(67), EECOC ), ( A(68), ELOADH )
3062          EQUIVALENCE ( A(69), ELOADC ), ( A(70), EREH ), ( A(71), EREC )
3063          EQUIVALENCE ( A(90), SECOU ), ( A(91), TZMAX ), ( A(92), TZMIN )
3064          EQUIVALENCE ( A(93), RHAMAX ), ( A(94), RHAMIN )
3065 C
3066          WRITE(CLU,7) HVFILE
3067          WRITE(CLU,8) USFILE
3068          WRITE(CLU,9) CLFILE
3069          WRITE(CLU,10) SHFILE
3070          WRITE(CLU,11) INFILE
3071          WRITE(CLU,12) SEASON
3072 C
3073          WRITE (CLU,1)

```



```

3074 WRITE (CLU,2) EFANS
3075 WRITE (CLU,1)
3076 WRITE (CLU,3)
3077 WRITE (CLU,1)
3078 WRITE (CLU,4) EECOH, -EECOG
3079 WRITE (CLU,1)
3080 WRITE (CLU,5) ELOADH, -ELOADC
3081 WRITE (CLU,1)
3082 WRITE (CLU,6) EREH, -EREC
3083
C
3084 WRITE (CLU,13) SECOUT/60.
3085 WRITE (CLU,14) TZMAX, RHAMAX
3086 WRITE (CLU,15) TZMIN, RHAMIN
3087
C
3088 RETURN
3089 1 FORMAT(/,1X,18X,4('-', '+', 2(16('-', '+')))
3090 2 FORMAT(1X, 'FAN ENERGY' ,1',G12.5, ' KWHI')
3091 3 FORMAT(1X, 22X, ' HEATING' | ' COOLING' |)
3092 4 FORMAT(1X, 'ECONOMIZER ENERGY' ,1', 2(G12.5, ' BTUI'))
3093 5 FORMAT(1X, 'LOAD ENERGY' ,1', 2(G12.5, ' BTUI'))
3094 6 FORMAT(1X, 'REHEAT/RECOOL ENERGY' ,1', 2(G12.5, ' BTUI'))
3095 7 FORMAT(1X, 'HVAC PARAMETERS FROM: ', A15)
3096 8 FORMAT(1X, 'USE PARAMETERS FROM: ', A15)
3097 9 FORMAT(1X, 'CLIMATE PARAMETERS FROM: ', A15)
3098 10 FORMAT(1X, 'SHELL PARAMETERS FROM: ', A15)
3099 11 FORMAT(1X, 'INITIALIZING DATA FROM: ', A15)
3100 12 FORMAT(1X, 'WEATHER SEASON USED: ', A15)
3101 13 FORMAT(/,1X, 'MINUTES OUTSIDE COMFORT ENVELOPE = ', F10.2)
3102 14 FORMAT(1X, 'MAXIMUM ZONE TEMPERATURE = ', F10.2, ' (', F5.1, '%RH)')
3103 15 FORMAT(1X, 'MINIMUM ZONE TEMPERATURE = ', F10.2, ' (', F5.1, '%RH)')
3104 END
3105 C=====
3106 SUBROUTINE SHOTS
3107 C=====
3108 C This is a special subroutine used for development of the emulator model
3109 C which will eventually be eliminated. Its purpose is to allow snapshots
3110 C of the state vector to be taken and stored in a file for analysis.
3111 C Instructions on which state vector variables are to be shot and what
3112 C interval the snapshots are to be taken at, are read from the file
3113 C assigned to logical unit 7. The snapshots are stored in the file
3114 C assigned to logical unit 10.
3115 C
3116 INTEGER DAY, HOUR, MINUTE, SECOND, ENDDAY, ENDHR, ENDMIN, ENDSEC
3117 COMMON / TIME / DAY, HOUR, MINUTE, SECOND, ENDDAY, ENDHR, ENDMIN, ENDSEC
3118 LOGICAL L
3119 COMMON / STATE1 / A(100), L(100)
3120
C
3121 LOGICAL NOINIT
3122 INTEGER IVECP(21)
3123 DATA NOINIT /.TRUE./
3124
C
3125 IF(NOINIT) THEN
3126 READ(7, *, ERR=900, END=900) REPINV, NCHAN, (IVECP(I), I=1, NCHAN)
3127 IF(NCHAN.GT.21)
3128 & STOP 'TOO MANY STATE VECTOR POSITIONS IN SNAPSHOT'
3129 GO TO 1000
3130 900 WRITE(1, FMT='(1X, "REPORT INTERVAL IN TIMESTEPS?")')
3131 READ(1, *) REPINV

```

```

3132         WRITE(1,FMT='(1X,"NUMBER OF VECTOR POSITIONS TO REPORT?")')
3133         READ(1,*) NCHAN
3134         WRITE(1,FMT='(1X,"POSITION INDICES?")')
3135         READ(1,*) (IVECP(I),I=1,NCHAN)
3136     1000     REPLEF = 1
3137             NOINIT = .FALSE.
3138         ENDIF
3139         REPLEF = REPLEF - 1
3140         IF(REPLEF.LE.0) THEN
3141             RTIME = FLOAT(DAY-1)*24. + HOUR +
3142             &   FLOAT(MINUTE)/60. + FLOAT(SECOND)/3600.
3143             WRITE(10,1)RTIME,(A(IVECP(I)),I=1,NCHAN)
3144             1 FORMAT(1X,011.5,6011.3:./,(1X,7011.3))
3145             REPLEF = REPLEF + 1
3146         ENDIF
3147         RETURN
3148     END
3149 C Version 1.5 - FEBRUARY 4, 1985 - W.B. MAY ,NATIONAL BUREAU OF STANDARDS
3150 C=====
3151     FUNCTION CPAIR(W)
3152 C=====
3153 C This function returns the specific heat of moist air as a function of
3154 C the humidity ratio.
3155 C
3156 C UNITS: specific heat is returned in Btu/lbm F
3157 C
3158 C REFERENCE: ASHRAE Handbook of Fundamentals 1981, page 5.3
3159 C
3160     CPAIR = 0.24 + 0.444 * W
3161 C
3162     RETURN
3163     END
3164 C=====
3165     FUNCTION RHOAIR(W,TF)
3166 C=====
3167 C This function returns the density of moist air as a function of
3168 C the humidity ratio and the dry bulb temperature.
3169 C
3170 C UNITS: density is returned in lbf/ft^3
3171 C         temperature TF must be in degrees F
3172 C
3173 C REFERENCE: ASHRAE Handbook of Fundamentals 1981, page 5.3
3174 C
3175 C PRESSR is atmospheric pressure in lbf/ft^2 (assumed standard atmosphere)
3176 C RAIR is the gas constant for dry air in ft-lbf/lbm R
3177 C TA is the absolute temperature of the air in degrees R
3178 C FTOR is the conversion factor from F to R degrees
3179 C
3180     PARAMETER (PRESSR = 2116.2, RAIR = 53.352, FTOR = 459.67)
3181 C
3182     TA = TF + FTOR
3183     RHOAIR = PRESSR / (RAIR * TA * (1 - 1.6078 * W) )
3184     RETURN
3185     END
3186 C=====
3187     FUNCTION ENTHAL(TDB,WPPDA,PA)
3188 C=====
3189 C from ASHRAE 1981 Fundamentals handbook, Page 5.4

```

```

3190 C
3191 C TDB = average dry bulb temperature in degrees F
3192 C PA = atmospheric pressure in inches of mercury
3193 C WPPDA = humidity ratio in pounds of water per pound of dry air
3194 C ENTHAL = enthalpy of mixture in BTU per pound of dry air
3195 C
3196 ENTHAL = 0.240 * TDB + WPPDA * (1061. + 0.444*TDB)
3197 RETURN
3198 END
3199 C=====
3200 FUNCTION TEMP(ENTHAL,WPPDA,PA)
3201 C=====
3202 C Based on equation for anthalpy in ASHRAE 1981 Fundamentals
3203 C handbook, Page 5.4, solved for TEMP.
3204 C
3205 C TEMP = average dry bulb temperature in degrees F
3206 C PA = atmospheric pressure in inches of mercury
3207 C WPPDA = humidity ratio in pounds of water per pound of dry air
3208 C ENTHAL = enthalpy of mixture in BTU per pound of dry air
3209 C
3210 TEMP = (ENTHAL - 1061.*WPPDA) / (0.240 + 0.444*WPPDA)
3211 RETURN
3212 END
3213 C=====
3214 FUNCTION DEWPT(WPPDA,PA)
3215 C=====
3216 C from ASHRAE 1981 Fundamentals handbook, Page 5.4, equation 40b
3217 C equation is valid in the range of TDB from 32 to 150 F.
3218 C
3219 C PA = atmospheric pressure in inches of mercury
3220 C WPPDA = humidity ratio in pounds of water per pound of dry air
3221 C DEWPT = dewpoint temperature of mixture in degrees F.
3222 C PW = water vapor partial pressure for the mixture in inches Hg.
3223 C
3224 PW = PA*WPPDA / (0.62198 + WPPDA)
3225 ALPHA = ALOG(PW)
3226 DEWPT = 79.047 + 30.5790*ALPHA + 1.8893*ALPHA*ALPHA
3227 RETURN
3228 END
3229 C=====
3230 FUNCTION HUMRAT(TDB,RH,PA)
3231 C=====
3232 C from ASHRAE 1981 Fundamentals handbook, Page 5.4
3233 C
3234 C TDB = average dry bulb temperature in degrees F
3235 C PA = atmospheric pressure in inches of mercury
3236 C RH = relative humidity in percent
3237 C HUMRAT = humidity ratio in pounds of water per pound of dry air
3238 C
3239 PASCAL = PA * 3.38638E+3
3240 TDBK = (((TDB-32)/1.8)+273.15)
3241 PWAT = RH * PWSAT(TDBK) / 100.
3242 HUMRAT = 0.62198 * ( PWAT / (PASCAL - PWAT) )
3243 RETURN
3244 END
3245 C=====
3246 FUNCTION RELHUM(TDB,WAIR,PA)
3247 C=====

```

```

3248 C from ASHRAE 1981 Fundamentals handbook, Page 5.4
3249 C
3250 C TDB = average dry bulb temperature in degrees F
3251 C PA = atmospheric pressure in inches of mercury
3252 C RELHUM = relative humidity in percent
3253 C WAIR = humidity ratio in pounds of water per pound of dry air
3254 C
3255 PASCAL = PA * 3.38638E+3
3256 TDBK = (((TDB-32)/1.8)+273.15)
3257 PWAT = PASCAL * WAIR / (.62198 + WAIR)
3258 RELHUM = PWAT / PWSAT(TDBK) * 100.
3259 RETURN
3260 END
3261 C=====
3262 FUNCTION PWSAT(T)
3263 C=====
3264 C from ASHRAE 1981 Fundamentals handbook, Page 5.2
3265 C
3266 C TDB = dry bulb temperature in degrees K.
3267 C PWSAT = saturation pressure of water vapor in pascals
3268 C
3269 DATA C1/-5674.5359 /, C2/6.3925247 /, C3/-0.9677843E-2 /
3270 DATA C4/ 0.62215701E-6/, C5/0.20747825E-8/, C6/-0.9484024E-12/
3271 DATA C7/ 4.1635019 /, C8/-5800.2206 /, C9/ 1.3914993 /
3272 DATA C10/ -0.04860239 /, C11/0.41764768E-4/, C12/-0.14452093E-7/
3273 DATA C13/6.5459673/
3274 T2=T*T
3275 T3=T2*T
3276 T4=T3*T
3277 IF(T.GT.273.15)GO TO 100
3278 PWSLN = C1/T+C2+C3*T+C4*T2+C5*T3+C6*T4+C7*ALOG(T)
3279 GO TO 200
3280 100 PWSLN = C8/T+C9+C10*T+C11*T2+C12*T3+C13*ALOG(T)
3281 200 PWSAT=EXP(PWSLN)
3282 RETURN
3283 END
3284 C=====
3285 FUNCTION TDBSAT(HSAT,PA)
3286 C=====
3287 C This function uses a table look-up using data from ASHRAE 1981
3288 C Fundamentals handbook, Page 6.5, table 2.
3289 C function is valid in the range of TDB from 32 to 100 F.
3290 C
3291 C TDBSAT = dry bulb temperature in degrees F at saturated conditions
3292 C PA = atmospheric pressure in inches of mercury assumed to be 29.921 inhg
3293 C HSAT = enthalpy of saturated mixture in BTU per pound of dry air
3294 C
3295 REAL H(69),T(69)
3296 DATA H / 11.760,12.170,12.587,13.010,13.441,13.878,14.322,14.773,
3297 & 15.233,15.700,16.175,16.660,17.152,17.653,18.164,18.685,
3298 & 19.215,19.756,20.306,20.868,21.441,22.025,22.621,23.229,
3299 & 23.850,24.484,25.131,25.792,26.467,27.157,27.862,28.582,
3300 & 29.318,30.071,30.840,31.626,32.431,33.254,34.097,34.959,
3301 & 35.841,36.743,37.668,38.615,39.583,40.576,41.592,42.633,
3302 & 43.701,44.794,45.913,47.062,48.238,49.445,50.681,51.949,
3303 & 53.250,54.582,55.951,57.355,58.794,60.271,61.787,63.343,
3304 & 64.940,66.578,68.260,69.988,71.761/
3305 DATA T / 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0,

```

```

3306      &      40.0, 41.0, 42.0, 43.0, 44.0, 45.0, 46.0, 47.0,
3307      &      48.0, 49.0, 50.0, 51.0, 52.0, 53.0, 54.0, 55.0,
3308      &      56.0, 57.0, 58.0, 59.0, 60.0, 61.0, 62.0, 63.0,
3309      &      64.0, 65.0, 66.0, 67.0, 68.0, 69.0, 70.0, 71.0,
3310      &      72.0, 73.0, 74.0, 75.0, 76.0, 77.0, 78.0, 79.0,
3311      &      80.0, 81.0, 82.0, 83.0, 84.0, 85.0, 86.0, 87.0,
3312      &      88.0, 89.0, 90.0, 91.0, 92.0, 93.0, 94.0, 95.0,
3313      &      96.0, 97.0, 98.0, 99.0, 100.0/
3314      C
3315      TDBSAT = TERPOL(T,H,HSAT,69)
3316      RETURN
3317      END
3318      C-----
3319      FUNCTION TERPOL(Y,X,XI,LTAB)
3320      C-----
3321      C This is a general function to interpolate a value from a table.
3322      C XI is the independent value, X is a list of independent values
3323      C Y is a list of dependent values. LTAB is the length of the table.
3324      C
3325      REAL Y(*),X(*)
3326      C
3327      C-----
3328      C Locate position in table
3329      C
3330      DO 1000 I = 1,LTAB
3331      IF((XI.GT.X(I).AND.XI.LT.X(I+1)).OR.
3332      & (XI.LT.X(I).AND.XI.GT.X(I+1))) GO TO 2000
3333      1000 CONTINUE
3334      C
3335      C-----
3336      C independent value supplied is not in table
3337      C
3338      IF(X(LTAB).GT.X(1)) THEN
3339      IF(XI.LT.X(1)) THEN
3340      TERPOL = Y(1)
3341      ELSE IF(XI.GT.X(LTAB)) THEN
3342      TERPOL = Y(LTAB)
3343      ENDIF
3344      ELSE IF(X(1).GT.X(LTAB)) THEN
3345      IF(XI.GT.X(1)) THEN
3346      TERPOL = Y(1)
3347      ELSE IF(XI.LT.X(LTAB)) THEN
3348      TERPOL = Y(LTAB)
3349      ENDIF
3350      ELSE
3351      TERPOL = 0.0
3352      ENDIF
3353      RETURN
3354      C
3355      C-----
3356      C interpolate
3357      C
3358      2000 TERPOL = (XI-X(1)) / (X(I+1)-X(1)) * (Y(I+1)-Y(1)) + Y(1)
3359      RETURN
3360      END

```

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET (See instructions)		1. PUBLICATION OR REPORT NO. NBSIR 85-3291	2. Performing Organ. Report No.	3. Publication Date JANUARY 1986
4. TITLE AND SUBTITLE BUILDING EMULATION COMPUTER PROGRAM FOR TESTING OF ENERGY MANAGEMENT AND CONTROL SYSTEM ALGORITHMS				
5. AUTHOR(S) William B. May, Jr. and Cheol Park				
6. PERFORMING ORGANIZATION (If joint or other than NBS, see instructions) NATIONAL BUREAU OF STANDARDS DEPARTMENT OF COMMERCE WASHINGTON, D.C. 20234			7. Contract/Grant No.	8. Type of Report & Period Covered
9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (Street, City, State, ZIP) Naval Civil Engineering Laboratory Office of Building and Community Systems U.S. Department of Defense U.S. Department of Energy Port Hueneme, California 93043 1000 Independence Avenue, SW Washington, DC 20585				
10. SUPPLEMENTARY NOTES <input type="checkbox"/> Document describes a computer program; SF-185, FIPS Software Summary, is attached.				
11. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here) A building emulator can be used to test energy management and control systems (EMCS). The emulator uses a computer program to simulate the responses of a building including the equipment, building space, and building envelope to EMCS commands. Building model software for the emulator has been developed at the National Bureau of Standards (NBS) in an effort to assist the United States Naval Civil Engineering Laboratory (NCEL), which is developing a sophisticated building emulator. The concept of the building emulator and the building emulator computer program are described in this report. The program includes the weather, the air handling unit, the zone, and the comfort model. In addition, the energy compilation routine is also included. The models presented here are simplified models. With these abridged models, a single zone building with exterior walls and a single deck air handling unit are simulated. A complete FORTRAN source code of the building emulator computer program is appended.				
12. KEY WORDS (Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons) air handling unit simulation model; building emulator; building space zone model; computer simulation; EMCS algorithms; energy management and control systems (EMCS); HVAC emulator; local equipment simulation model; testing EMCS algorithms.				
13. AVAILABILITY <input checked="" type="checkbox"/> Unlimited <input type="checkbox"/> For Official Distribution. Do Not Release to NTIS <input type="checkbox"/> Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402. <input checked="" type="checkbox"/> Order From National Technical Information Service (NTIS), Springfield, VA. 22161			14. NO. OF PRINTED PAGES 132 15. Price \$16.95	

