AN ABSTRACT OF THE THESIS OF

Stephen John Fox for the degree of Master of Science in Mechanical Engineering presented April 26, 1984

Title: Computer Simulation of a Combined-Cycle Biomass-Fueled Power Plant

Abstract approved: Redacted for Privacy

Gordon M. Reistad

This thesis presents thermodynamic evaluations of various biomass-fueled combined-cycle power plants using a computer model developed for simulating such plants. The combined-cycle systems investigated are characterized by having a wood-fueled combustor, an indirect-fired gas turbine, and a steam cycle. This work was part of a larger biomass energy project at Oregon State University.

The systems evaluated consisted of a "base case" design and five variations thereof. The base case itself includes a fuel preparation section with fuel drying, a combustor, an indirect-fired gas turbine system, a waste heat recovery boiler, and a Rankine cycle steam system. An important facet of this design is that hot combustion gases as well as the exhaust gases from the waste heat recovery boiler are used to dry the fuel. Variations of the base case that were investigated included (i) elimination of the use of hot combustion gases for drying, (ii) steam injection into the gas turbine, (iii) boosting the indirect-fired gas turbine inlet

temperature with a trimburner, (iv) eliminating fuel drying, and (v) a combination of (iii) and (iv).

The simulation model developed consists of an main program and component models. The main program solves the simultaneous nonlinear algebraic equations that arise in modeling steady state thermal systems. The equations of the process in each component are evaluated in the respective component model. The unknown variables require initial guesses at the start of simulation and are solved by iteration using successive substitution until convergence is reached.

The results of the simulations allowed the various systems to be compared with regard to efficiency, power output, total heat transfer area, and irreversibility. The various systems have similar net efficiencies, although those using a trimburner have a net efficiency about 2% higher than the others. In fact, the efficiency of the fossil fuel used in the trimburner is about 42%. The power output of the various systems, like the efficiency, is about the same except for those that utilize the trimburner. All of the systems without the trimburner have essentially the same overall irreversibility. But some of the systems have very large mixing irreversibilities and others have greater irreversibilities in the heat exchanger. The occurance of the irreversibilities in the heat exchangers corresponds with smaller heat transfer areas. Total heat transfer area changed by as much as 6000 square feet (30%) from one system to another. These thermal analysis results show that detailed economic evaluations are necessary to fully determine the relative desirability of the systems.

Computer Simulation

of a Combined-Cycle Biomass-Fueled Power Plant

by

Stephen John Fox

A THESIS

Submitted to

Oregon State University

In partial fulfillment of
the requirements for the
degree of

MASTER OF SCIENCE

Completed April 26, 1984

Commencement June 1984

APPROVED:

Redacted for Privacy

Professor of Mechanical Engineering in charge of major

Redacted for Privacy

Head of Mechanical Engineering Department

Redacted for Privacy

Dean of Graduate School

Date thesis is presented April 26, 1984

Typed by Stephen John Fox

ACKNOWLEDGMENTS

This project was not a singular effort; many were involved that deserve recognition. Without all of their help, the work could not have been accomplished.

Dr. Gordon M. Reistad has introduced me to several quality projects, both as an undergraduate and a graduate student. His dedication and prompt assistance have been crucial to the completion of the projects, especially this thesis. He has helped me a lot through the past eight years in many ways.

Tim Bauer and **Siggi Brynjolfsson** wrote most of the component subroutines for the biomass plant, without which the simulations could not have been done. In addition to their technical input, their humor kept the project going during the tougher periods.

I would like to also thank **Randy Keller** and **John Martin**, who did the drafting on this project, **Dr. Dwight Bushnell**, for his help in the overall biomass project, **Dr. Tim Kennedy**, for assistance on some of the numerical problems, and **Randy Jacobson**, the graduate school representative.

The U. S. Department of Agriculture funded this project.

Finally, a special thank you to my wife, **Lori**, who helped a great deal in the editing of this thesis, as well as putting up with me these last two years. She sacrificed a great deal when we decided to return to school, and her understanding and dedication are very appreciated.

TABLE OF CONTENTS

LIST OF FIGURES

NOMENCLATURE

General Notation

A... -- area
c... -- a heat exchanger coefficient
CFM... -- volumetric flow rate, cu. ft/min
dP... -- delta P (pressure)
EFF... -- efficiency
fou... -- fouling in heat exchanger
fs... -- factor of safety
HHV... -- higher heating value
HR... -- heating rate
ID... -- inner diameter of pipes
IRR... -- irreversibility
Kp... -- conductivity of the pipe
len... -- length
LHV... -- lower heating value
M... -- molecular weight
MR... -- mass flow rate
npn... -- no. of pipes normal to flow dir
npp... -- no. of pipes parallel to flow dir
nu... -- no. of "U"'s in heat exchanger
OD... -- outer diameter of pipes
P... -- pressure
Q... -- heating rate (should be HR...)
rad... -- radiation loss ratio
ruf... -- relative roughness in piping
SnD... -- normalized spacing of tubes, normal to flow dir
SpD... -- normalized spacing of tubes, parallel to flow dir
T... -- temperature
typ... -- type of heat exchanger (1, 2, ...: see hx sub for details)
U... -- overall heat transfer coefficient
V... -- velocity
W... -- specific humidity (mass ratio water/total)
WR... -- work rate
x... -- a heat exchanger coefficient
X... -- ratio (steam quality, etc.)
y... -- molar ratio

...a.. -- air
...b.. -- biomass fuel
...c.. -- combustion products
...e.. -- exhaust
...s.. -- steam and water

...ap. -- air pollution devices
...aux -- auxiliary
...bag -- baghouse

...blr -- boiler
...fn. -- fans
...cyc -- cyclone
...cmb -- combustor
...cmp -- compressor
...cnd -- condenser
...cnv -- conveyor
...col -- mechanical collector
...ctr -- cooling tower
...cyc -- cyclone
...ds -- dead state (for 2nd law analysis)
...dry -- dryer
..econ -- economizer
...gb1 -- gear box connected to gas turbine
...gb2 -- gear box connected to steam turbine
...gn1 -- generator connected to gas turbine
...gn2 -- generator connected to steam turbine
...grs -- gross
...gtb -- gas turbine
...hml -- hammer mill
...hx1 -- heat exchanger, steam-gas
...hx2 -- heat exchanger, air-gas
...inj -- steam injector
...mxc -- mixer just after the combustor
...mxd -- mixer before the dryer
...net -- net (gross - auxiliary)
...pmp -- pump
...scw -- screw (fuel feeder)
...st. -- steam turbine
...stp -- standard temperature and pressure
.super -- superheater
...trm -- trimburner
....CD -- carbon dioxide (CO2)
....W -- water (H2O)
....O2 -- oxygen
....N2 -- nitrogen
....Ar -- Argon
....CO -- carbon monoxide

.....a -- air side
.....g -- gas side
.....w -- water side

.....m -- minimum allowed

Specific Identifiers

AFrat -- air/fuel ratio

```
cmpbld -- percent bleed air from compressor
dPblrg -- delta P of gas in boiler
dPblrw -- delta P of water in boiler
dPhx1g -- delta P of gas   side of gas-water heat exchanger
dPhx1w -- delta P of water side of gas-water heat exchanger
dPhx2a -- delta P of air side in the gas-air heat exchanger
dPhx2g -- delta P of gas side in the gas-air heat exchanger
drylos -- fraction of available heat loss in dryer
EFFHg  -- gross first law efficiency, based on HHV
EFFHn  -- net   first law efficiency, based on HHV
exair  -- excess air
HHV    -- higher heating value of the fuel
HRcnd  -- heating rate of condenser (actually a 'cooling' rate!)
HRloss -- heating rate loss from combustor
MR...m -- minimum flow rate allowed in stream "..."; if less, set 0.
MRash  -- mass flow rate of ash
MRbdwd -- mass flow rate of bone dry wood
MRcnd  -- mass flow rate of condenser cooling fluid
MRdirt -- mass flow rate of dirt
MRstp  -- mass rate through compressor at standard T and P
Prcmp  -- pressure ratio through compressor
Prcinj -- percent injection of steam into gas turbine
radlos -- radiation loss in combustor
Tsuper -- superheating in boiler
unbcbn -- ratio of unburned carbon
XCCO   -- fraction of carbon burned to carbon monoxide
XC,XH2,XN2,XO2,Xdirt,Xash -- mass ratio of carbon, hydrogen, nitrogen,
                             oxygen, dirt, and ash in the fuel
```

COMPUTER SIMULATION
OF A COMBINED-CYCLE BIOMASS-FUELED POWER PLANT

Chapter 1
INTRODUCTION

The objective of this work was to evaluate the thermodynamic performance of selected biomass fueled power plants for the generation of electricity by computer simulation modeling. This work was part of a larger project at Oregon State University to study the biomass fueled power plant concept and to make recommendations on an optimal system configuration.

The power plant studied in this thesis is a combined-cycle plant of nominal 10 MW capacity that burns wood fuels obtained from forest fuel crops, forest residues, and/or industrial mill residues. This fuel is to be trucked to the power plant from various sites in the vicinity. Furthermore, the system is intended to be semi-portable because biomass fuel supplies are often limited in volume and availability.

The primary components of the combined cycle as studied consist of an indirect fired gas turbine and a waste heat boiler used in a conventional Rankine cycle steam turbine system. Although a variety of system designs were modeled, they all are deviations from a base case design. Figure 1.1 shows a schematic of the various flows among the components of the base case plant (the specifics of the different systems studied are presented in Chapter 4). There are five types of flows depicted in Figure 1.1: (i) biomass fuel, using state point numbers beginning with "B", (ii) air, using the letter "A", (iii) combustion products (flue gas), using the letter "C", (iv) steam, using the letter "S", and (v) exhaust gases, using the letter "E". Table 1.1 lists the nomenclature of the equipment components in Figure 1.1.

The wet fuel is broken into 1/2 inch or smaller particles by a hammermill, and then partially dried in a triple-pass rotary drum dryer. The dried fuel is then passed through a cyclone to separate the fuel from the drying gases and another hammermill to reduce its size further. The fuel is stored in a bin until fed via a screw feeder into the combustor. Meanwhile, air is compressed and then passed through an air-gas heat exchanger. This high temperature and pressure air is expanded through a gas turbine to produce electricity through an attached generator, and is then sent to the combustor. The combustor inlet air has a fairly high temperature, and the combustion process occurs with a high level of excess air. The resulting flue gases are passed through a cyclone to remove particulates and then fed to the steam superheater at state point C12, and the air-gas heat exchanger at state point C13. The flue gases are further cooled in a waste heat boiler, and then passed through the dryer and exhausted to the atmosphere. Along the way the

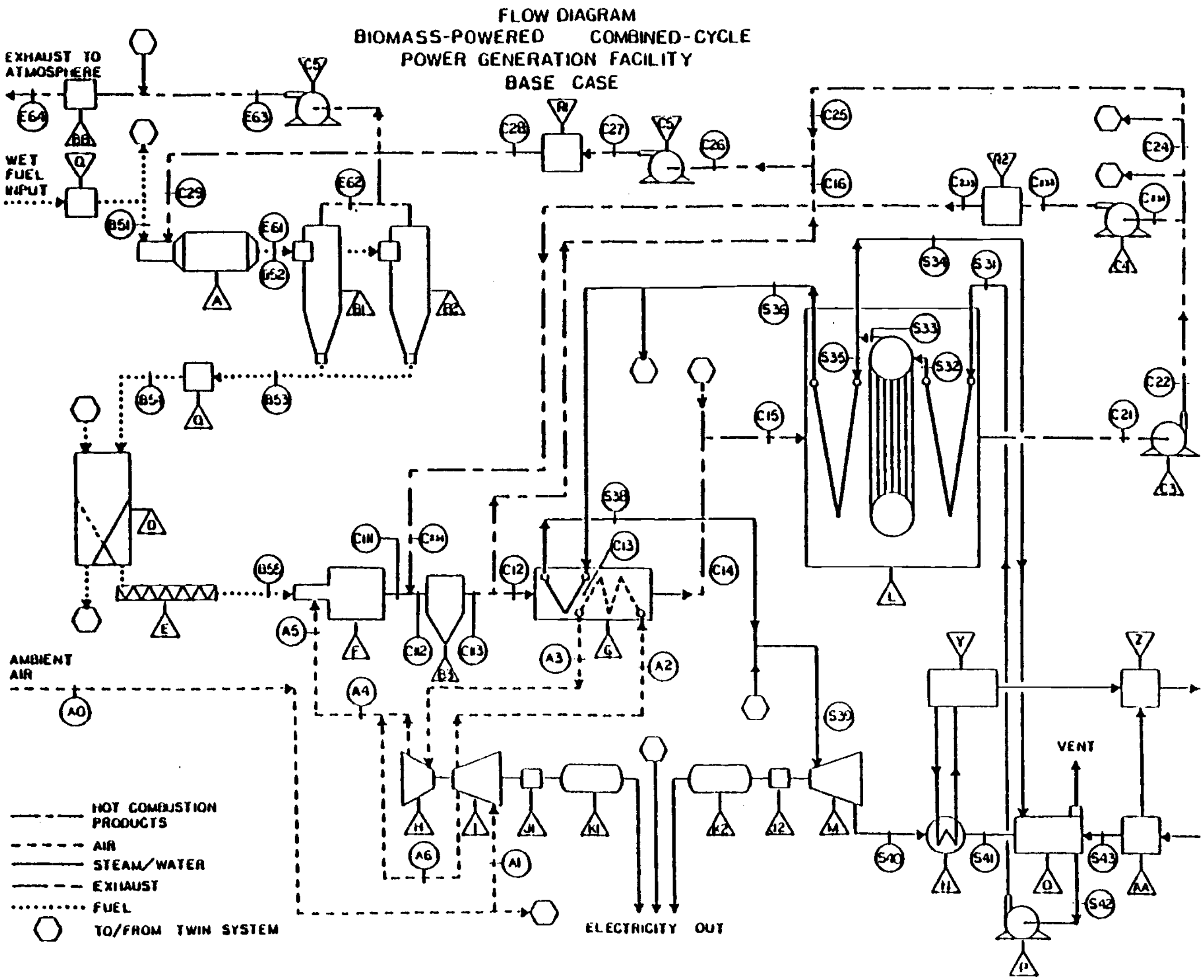


Figure 1.1 Schematic diagram of base case power plant

TABLE 1.1 EQUIPMENT COMPONENTS IN BASE-CASE DESIGN
(Letters are identified on figure 1.1)

Dryer - A

Cyclones - B1, B2 and B3

Fans - C1, C2, C3, C4 and C5

Storage Bins - D

Screw Feeders - E

Combustor - F

Heat Exchanger (steam/flue gas and air/flue gas) - G

Gas Turbines - H

Compressor - I

Gearboxes - J1 and J2

Generators - K1 and K2

Waste Heat Boiler - L

Steam Turbine - M

Condenser - N

Deaerator - O

Pump - P

Hammer Mill - Q

Air Pollution Control Devices - R

gases are passed through several pollution control devices.

All of the above mentioned equipment is duplicated in a twin system, except for the waste heat boiler. The flue gas from the two systems is combined at state point C15 and fed to one waste heat boiler. This boiler consists of an economizer section, a boiler, and a primary superheater. The steam is further superheated in the steam superheater at state point S38. It is expanded through a steam turbine, which produces additional electricity through its own generator. The steam is condensed and pumped back to the waste heat boiler.

Part of the flue gases exiting the waste heat boiler are recycled back to the combustor at state point C231, in order to keep the combustor exit temperature within the limits demanded by the heat exchanger. In addition, some of the hot flue gases are routed directly to the dryer through state point C16.

Chapters 2 and 3 discuss the power plant computer simulation. Chapter 2 discusses the general methods of simulation, while Chapter 3 is concerned with the specific computer program developed here. Chapter 4 describes the various power plant systems and presents the results and conclusions from the simulations performed.

Chapter 2

THERMAL PROCESS SIMULATION

There are many programs available for simulation of specialized types of thermal systems. The question then arises: why not use a program that is already available? This question was investigated as part of the overall biomass system project. It was decided by the biomass team members to develop a simulation program rather than use one of those already available for a number of reasons, some of which were:

- No commercially available code was identified that would work for all of the components that were in the biomass power plant. The user of a commercial code is not generally allowed to study and modify the codes to make them work for the components that aren't already part of the code.

- Direct purchase of proprietary codes is very expensive.

- Public domain codes that were investigated had poor documentation or were not applicable.

This chapter presents the "general approach" to modeling that was used to develop the simulation model for the biomass project. The specific program used in this thesis is presented in Chapter 3.

Modeling a large system consisting of many components is a difficult task to tackle. It is easier to divide the large system into smaller, more manageable, sections or modules. This is accomplished by creating a separate module for each component, (although similar components can often use the same mathematical model).

Assembling the components into a system model presents new problems when the calculation sequence becomes very complicated. The flow of information among the components becomes a major bookkeeping task. For many complicated systems the solution involves a set of nonlinear simultaneous equations as well.

Each of these ideas is discussed in detail below along with the concepts of component models, modules, recycle loops and information flow.

COMPONENT MODELS

Each component in a thermal system has its own mathematical model. The model includes such physical laws as the first and second laws of thermodynamics and continuity, as well as heat transfer relationships, experimental correlations, manufacturer's data, material property data, and economic data.

The physical laws form the basic model of most components. The first law and second law of thermodynamics coupled with the conservation of mass (continuity equation) are used to perform energy and mass balances of each component.

The thermodynamic data for the fluids are perhaps the most often needed material property data, and are a vital part of the modeling process. For computer programs, fitting the data to equations is the most convenient and compact form, as opposed to using tables. This can be a major task since the properties often need to cover a wide range. It may be difficult to accurately fit one equation to the data over such a range. In this case the data can be separated into several smaller ranges. Published curve fit equations can be found in the literature for some fluids. For example Zuck (1981) has compiled a set of equations for steam properties over an extensive range of temperatures and pressures. The equations from Zuck were used for this project (Fox, 1984).

Equations for flue gas properties need to account for a wide range of compositions. The gases can be broken up into each constituent. Using specific heat and other data for each constituent, the property is calculated for each constituent individually. These individual properties are then combined on a molar ratio basis to get the overall property value.

For components whose sizes have already been determined, actual equipment models, rather than theoretical models, are used for each component. These models may employ heat transfer relationships, manufacturer's data, experimental data, and materials data to calculate the outputs of the component.

Of course the size of the components affects the cost, and so economic data for the components may be included in a model. Comparisons of different sizes of the components along with fuel costs help determine the most economic system.

MODULES

Each type of equipment has its own component model. A thermodynamic process may have several pieces of the same type of equipment. Each has the same model, but the inputs and outputs for each use of that model are usually different. Each time a particular model is used it represents a new "module".

Often, there are groups of calculations that do not belong to a particular piece of equipment, such as overall system calculations of total cost, power output, efficiencies, power requirements, and others. This set of calculations would also be a "module".

Thus the term "module" refers to each set of related calculations which are either component models or other calculation sets. In the program, each subroutine call is a separate module.

RECYCLE LOOPS

Simple systems can be calculated one module at a time in an obvious sequence until all modules have been performed. The starting point is easily determined; it is the point where there is only one unknown value. This value is used to calculate the next unknown value. The calculations proceed sequentially through each module, passing through each module just once.

However, many thermal systems can not be attacked in such a straightforward manner. Figure 2.1 shows a more complicated system. A known temperature and flow rate enters the group of modules as stream 1. A required lower temperature is desired at stream 2 due to material properties in the cooler. To achieve this entrance temperature, a cooler stream 5 is mixed with stream 1. But the temperature of stream 5 depends on the flow rate through the cooler.

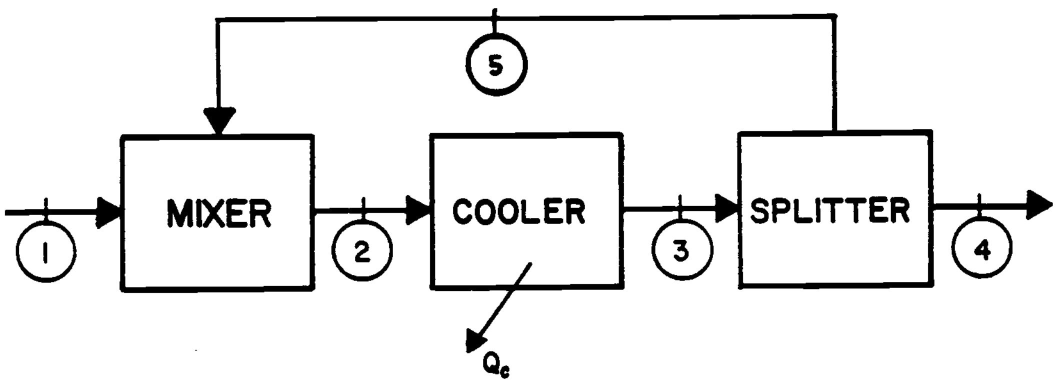


Figure 2.1 Recycle flows -- physical.

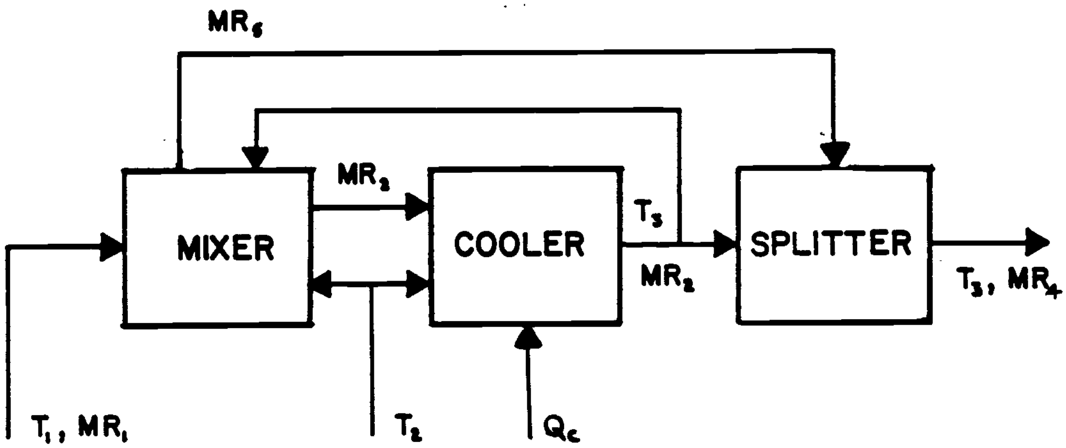


Figure 2.2 Recycle flows -- information.

There are three unknowns: the mass flow rates at streams 2 and 5, and the temperature at stream 3 (MR2, MR5, and T3). The two flow rates are related by continuity. If the value of one is known, the value of the other is calculated immediately. That leaves two unknowns, say MR2 and T3. If the value of MR2 was known, T3 could be calculated using an energy balance around the cooler module. On the other hand, if the value of T3 was known, the value of MR2 could be calculated using an energy balance around the mixer module.

The flow of information is shown in Figure 2.2. The value of information stream T3 needs to be guessed, used in the mixer module, then recalculated in the cooler. This is the recycle loop for information stream T3. In a larger system there may be many of these unknowns. The mathematical solution is to solve simultaneous equations for the unknowns.

INFORMATION FLOW DIAGRAMS

To organize the flow of information among the various modules, it is helpful to construct information flow diagrams. These diagrams graphically show the inputs and outputs of each module in a system. The inputs and outputs are not necessarily the physical inputs and output streams, but generally are any form of information needed to perform the calculations of the module, and the results of that module's calculations.

Figure 2.3 is an example of an information flow diagram for the process depicted in Figure 2.1. The inputs to each module enter at

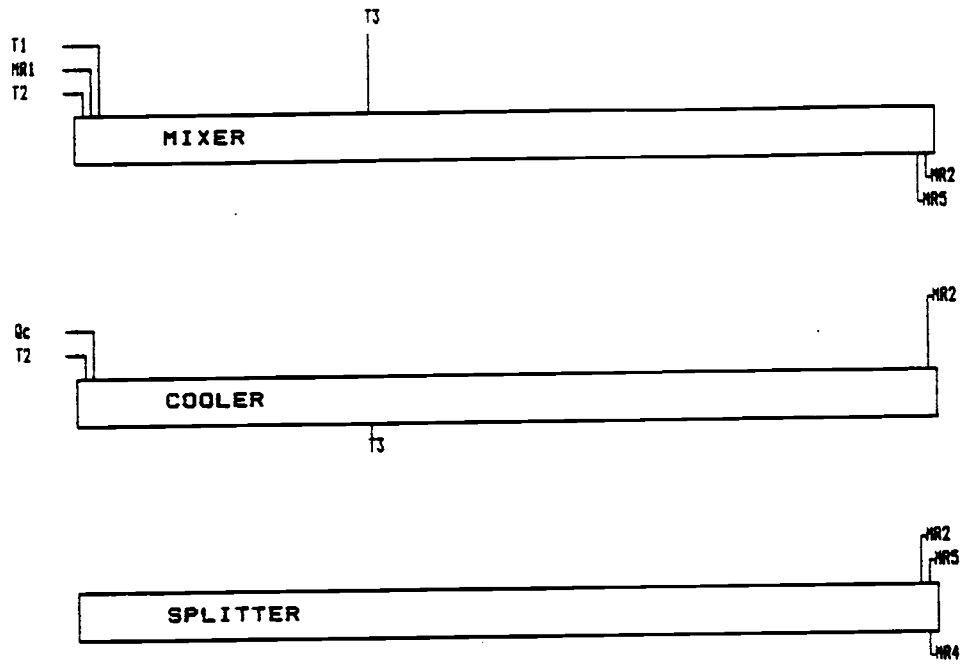


Figure 2.3 An example information flow diagram.

the top of the box. The outputs exit from the bottom of the box. This diagram will be discussed further in succeeding sections. A program was written to draw these diagrams, which has a limited amount of error checking to ensure a particular value is not calculated in more than one component module, and the input variables are output from another module. See Appendix A for details concerning its use and a listing of the program source code.

TYPES OF INFORMATION STREAM DATA

The information streams are the input and output data used in calling the modules. The inputs are of three types: constants, variables, and intermediate calculations. Outputs are also of three types: variables, intermediate calculations, and printed results.

Constants: a constant is information that is known before simulation begins and does not change for a particular simulation. Thermodynamic state points that are specified, such as the temperature and pressure of the atmosphere, are modeled as constants. Known component sizes and efficiencies are also treated as constants. In Figure 2.3, the constants are T1, MR1 and T2 for the mixer module, and Qc and T2 for the cooler module.

Occasionally, the value assigned to a constant may present an impossible physical situation. A method to rectify this is to change the constant's value in one of the modules. In this case, it is very important to check if any constants have been changed during the simulation. This method should be used sparingly, as it is hard to trace the change. It is better to be careful in assigning values to constants.

Variables: a variable is information that is unknown before simulation and must be determined during the simulation. It is evaluated from the calculations of the individual modules. A particular variable is an output from one module, as well as an input to another module. Each variable can be an input to several modules, but must be an output from only one module. Thus, variables are passed between modules.

Variables are necessary (rather than intermediate calculations, explained below) due to information recycle loops. An iteration through the modules of the system must be used to obtain the correct value of each variable. An initial guess for each variable is made before simulation begins. In Figure 2.3, the input variable is T3, input to the mixer. The cooler has T3 as an output variable.

Intermediate calculations: an intermediate calculation is similar to a variable, except that an initial guess of its value is not required. It must be output from one module before it can be an input to another downstream module(s). Its value must be calculated before it can be used, forcing a particular order of the modules that use it. In Figure 2.3, MR2 and MR5 are output intermediate calculations from the mixer; MR2 is an input intermediate calculation into the cooler; MR2 and MR5 are input intermediate calculations into the splitter; MR4 is output from the splitter.

Printed results: a printed result is used to denote a calculated piece of information that is not passed to another module. A module calculates its value and then simply prints it.

In summary, constants are known before simulation, and do not change during the simulation run. Variables are guessed before simulation, require iteration to determine their correct values, and are global outputs. Intermediate calculations do not require guesses, are not iterated on directly, require a certain order of the modules that use them, and are global outputs. Printed results are simply local outputs of a module.

SOLVING THE NONLINEAR SIMULTANEOUS EQUATIONS

In general, the equations concerned with thermodynamic processes are not linear, but are in some arbitrary algebraic form. Two popular methods for solving simultaneous non-linear equations are the successive substitution method and the Newton-Raphson method (Stoecker, 1980).

The successive substitution method starts the calculations at the first module, and proceeds through the remaining modules until all have been performed. The new values of each variable are then used to calculate each module again (they are successively substituted into a new iteration of calculations). This method is straightforward, easy to program on a computer, and uses a minimum amount of memory. It may converge slowly, or can diverge in some cases. However, for the processes used in this project, convergence always occurred, and the speed of convergence was sufficient.

The Newton-Raphson method is a second order iterative technique, so in many cases will converge faster than successive substitution. It uses a matrix of partial derivatives and a linear equation solver to obtain the new values for the variables. For the processes in this project, it is slower due to the extensive matrix manipulation, and it requires a large amount of storage; calculations of the partial derivatives (obtained numerically) are also time consuming.

After experimenting with these two methods, successive substitution was found to be the better simultaneous equation solution technique. Usually, not more than 15 iterations were required to converge to a 1% tolerance with up to 70 variables. With good guesses, (e.g. the values of the last run used as the initial guesses for the current simulation), convergence within 1% was often reached in less than 6 to 8 iterations.

Chapter 3

PROSIM, THE PROCESS SIMULATION PROGRAM

The computer program developed in this project to simulate a process is called PROSIM. The main subroutine of PROSIM is EQNS, which contains the calls to the various modules in the process. This chapter discusses PROSIM and EQNS, along with the inputs and outputs required. Following that, the steps needed to take a process from the initial specifications to the final simulation are outlined. An example of a simple system is presented in Appendix B.

THE STRUCTURE OF PROSIM

PROSIM is a general-purpose steady-state system simulation program that solves the simultaneous non-linear equations of the overall system model. It is used in conjunction with subroutine EQNS, which calls the various modules. Each module calculates some output values as functions of some input values. Before the simulation is performed, some of the inputs to a particular component model are known, and others are unknown (since they are outputs from other components).

Figure 3.1 shows the structure of PROSIM. PROSIM reads the input files with subroutines GETC, GETV and GETR. It then calls EQNS, which in turn calls the various modules in the thermal process. The modules may use property functions for the fluids involved.

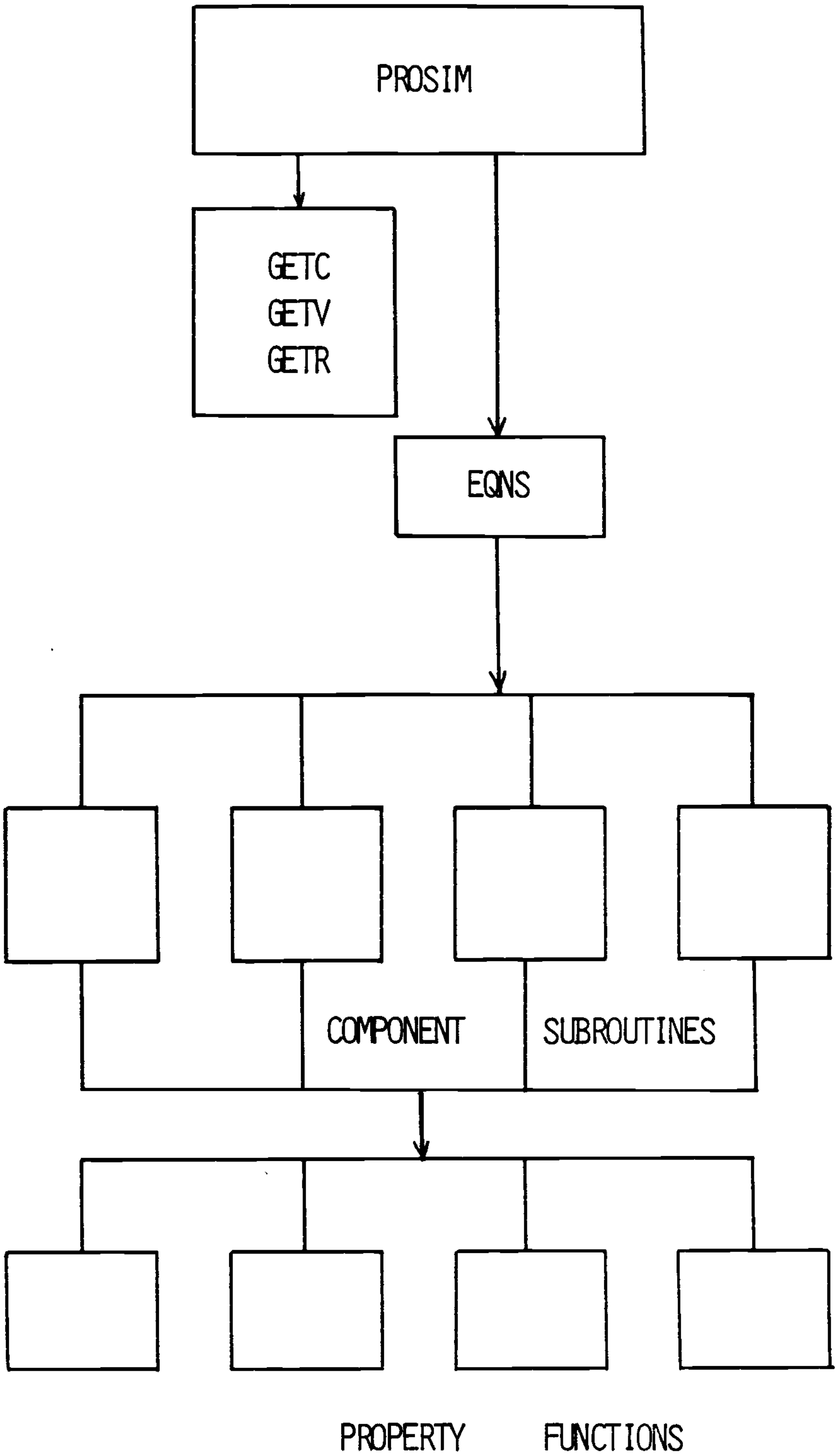


Figure 3.1 The structure of PROSIM.

PROSIM uses the successive substitution method to solve for the unknowns. Guesses must be made for each unknown, and the program iterates by calculating new values of the unknowns until a convergence criteria is satisfied. The results from EQNS are used again in another call to EQNS, until convergence has been reached. A listing of PROSIM (with GETC and GETV) and EQNS (with GETR) is in Appendix B.

Pseudocode of PROSIM

The pseudocode for PROSIM is listed below:

```
Initialization.

Obtain from the input files:
   Values of the constants
   Initial guesses and allowable bounds for the variables
   Names of the constants, guesses and intermediate
      calculations.


Repeat
   Calculate new variables and intermediate calculations,
      based on the values of the constants, and the previous
      values of the variables and intermediate calculations.
   Check for convergence of the variables.
   Check for maximum number of iterations.
   Check for divergence (out of bounds).
Until convergence, maximum iterations, or divergence.

Print the results.
```

Details concerning each part are presented in the next few sections.

Initialization

Initialization consists of several inputs the user must specify. These are (i) a debug flag (True/False), (ii) the output title, (iii) the maximum permissible tolerance, (iv) the maximum permissible number of iterations, and (v) five input and output file names.

calculated by the program.

Iteration to Solve for the Unknowns

Now the program is ready to perform the iteration process to solve for the variables and results. PROSIM calls EQNS, the subroutine that effectively contains all the equations of the system. New variables are calculated, and compared to the previous values (for the first time, they are compared with the initial guesses). Checks are made for convergence, maximum iterations, and divergence. When one of these three cases occurs, the program stops the iteration process.

The convergence criteria depends on the magnitude of the variable being checked. "Large" numbers have a magnitude bigger than the specified tolerance. Convergence is reached when the relative change is small, i.e. ratio

$$\text{abs}((v - v_{old}) / v) \quad < \text{tolerance.}$$

where: v is the current value of the unknown variable

v_{old} is the last value of the unknown variable

"Small" numbers have a magnitude less than the specified tolerance, and could conceivably equal zero. In this case, convergence is reached when the absolute change is small, i.e. when the difference

$$\text{abs}(v - v_{old}) \quad < \text{tolerance.}$$

Printing the Results

If the debug flag is on, the EQNS subroutine prints all printed

The debug flag signals the program to print extra information. When it is on, the value of each variable is printed to the console for each iteration, and each module executes its print statements. The output title is printed at the top of each page of the printout. The iteration tolerance is the maximum permissible tolerance mentioned in the convergence criteria section above. A typical value is 0.01. The maximum iterations is an upper limit on the number of iterations the program will allow. If convergence has not been reached, the program will print such a message.

Two input files names are required, one for the constants, and one for the variables. Three output files names are needed, one for warnings, one for the listing of the constants, variables, and intermediate calculations, and one for the component module outputs. These files are discussed in more detail in the "Data Input" and "Printing the Results" sections below.

Data Input

The input data are obtained from two input files. The first input file is for the constant names, their values, and their units. The second input file is for the variable names, their guessed values, their minimum and maximum allowable bounds, and their units.

The names and units for the intermediate calculations are not listed in a file. They are listed in subroutine GETR, which is physically located in the FORTRAN source code just after subroutine EQNS (see Appendix B for the listing). Of course, their values are information every iteration. If it is off, EQNS is called one more time after convergence is reached to print the output. (However, if the program diverges or reaches the maximum number of iterations, EQNS is not called again to provide the printed results.) Finally, the current values of the constants, variables, and results are printed. If the value of any constant is changed somehow during the iteration process, it's old and new values are reported.

The output is sent to three files: the warning file, the result file, and the print file. The following explains each file.

The **warning file** is where warnings from various routines are sent. For example, if the enthalpy function for water is given a temperature outside the accurate domain of its curve-fitting equation, a warning statement is printed. Usually, this file is assigned to the console.

The **result file** is where the final values of the constants, variables and results are sent. It is always created, whether convergence, divergence, or maximum iterations has occurred. Usually, it is sent to the printer, although some workers prefer to send it to a file for later use.

The **print file** is where output from the various component modules are sent. Most component modules print some local results that do not need to be passed to the main program. These results are sent to a separate output file than the global result file mentioned in the previous paragraph. The print file is also usually assigned to the printer.

THE STEPS INVOLVED IN MODELING WITH PROSIM

To use PROSIM, the system needs to be broken up, each section modeled, and then connected together to form the overall system model. This procedure for taking a system from the initial specification stage to the final simulation stage consists of several steps, shown in Figure 3.2. The steps are: (A specific example of a simple system is contained in Appendix B.)

1) Decide what values are constant, and what values are unknown.

The system's mathematical model involves many parameters and variables; it is necessary to place each value into one of these categories. In order to decide what values are required, a knowledge of how the components are modeled is also necessary. A simple model requires only a few inputs and outputs; a complex model may require a large number of inputs and outputs. Thermodynamic considerations are vital to these decisions. This step will often coincide with steps 2 and 3 for these reasons.

2) Determine the inputs and outputs for each component and draw the information flow diagrams.

This step, along with step 1, defines all the values required for the simulation. The inputs are separated into knowns and unknowns (input unknowns are actually outputs from other components). The knowns are the constants discussed in chapter 2, and the unknowns are the variables and intermediate calculations also discussed in

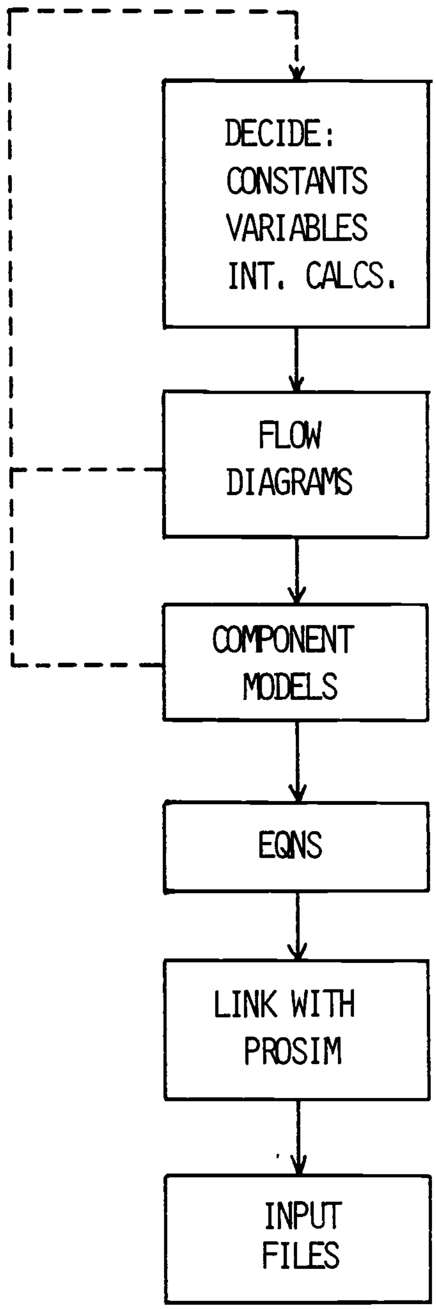


Figure 3.2 The steps in modeling with PROSIM.

that chapter. The outputs are variables, intermediate calculations, and printed results. Information flow diagrams show this pictorially.

3) Write the component subroutines.

The equations for each model can be determined knowing the inputs and outputs for a particular model. These equations, with the property functions, form the mathematical model. This step is integral with steps 1 and 2, since a knowledge of the model is required to determine the inputs and outputs of the model. Also, the capability of the models determines what values may be fixed as constants, and what may be considered unknowns for the simulation.

4) Write the EQNS subroutine.

Using the information flow diagrams as a tool, the subroutine call statements for each module is written in EQNS, with its corresponding inputs and outputs. The inputs are in one of the three arrays: C (constants), V (variables), and R (intermediate calculations). Printed results do not show up in the argument list for the subroutine calls; they are local to the particular component subroutine.

5) Link PROSIM, EQNS, and the component subroutines.

Using the appropriate software, the programs source codes are compiled and linked into one load module file (also known as a run file).

6) Prepare the input data files.

Two data files are needed: one for the constants, and one for the variables. The constants file contains the name, value, and units for each constant. The variables file contains the name, initial guess, minumum and maximum values allowed during iterations (to prevent the program from diverging without any indication of how it diverged), and units for each variable.

7) Run the simulation.

Once steps 1 through 5 are completed, different runs may be made by changing the files in step 6. The tolerance and maximum iterations are specified at run time, along with an output title. A debug option is available to have the program print all results during each iteration.

VERIFICATION OF PROSIM

A fairly simple textbook problem taken from Stoecker (1980) was solved with PROSIM to verify the solution technique used in PROSIM. The problem is presented in detail in appendix B. The solution was obtained in less than 10 iterations, even though the initial guesses were not very close to the correct values. PROSIM has also been compared with SYNTHA II for the Biomass project by Kennedy (1983). The SYNTHA II model was simplified because it does not have an adequate dryer model. However, good agreement in all the other components was obtained between the two programs.

Chapter 4

BIOMASS PLANT SIMULATIONS

The combined-cycle biomass-fueled power plant concept was originated by Biomass Energy Corporation of Portland, Oregon. Figure 1.1 shows the original configuration of the plant. It consists of three major sections: a fuel preparation area, an indirect fired gas turbine, and a Rankine cycle steam system. Wet fuel enters the dryer, is partially dried, then fed into the combustor. Meanwhile, air enters the compressor, is heated in the air-gas heat exchanger, expanded through the gas turbine, and also fed into the combustor. Some of the flue gas exiting the combustor is routed directly to the dryer. The rest is routed through the steam-gas and air-gas heat exchangers and the boiler. Some of the flue gas exiting the boiler is then recycled to the combustor to keep the heat exchanger inlet temperature within its allowable limit, and the remainder is routed to the fuel dryer. Bauer, et al (1983) discuss the plant in more detail, specifying the type of combustor, dryer, heat exchanger, gas turbine/compressor, and steam system.

This chapter presents a comparison through computer simulation of six biomass system alternatives to the configuration of Figure 1.1. First the systems are described and the results are presented. Several parametric evaluations were made on the most promising system. Finally, conclusions are presented.

It must be noted that in all of the computer simulations only half of the plant was modeled. The actual plant consists of twin fuel preparation areas, twin combustors, twin indirect fired gas turbine systems, and a single steam system. However, all Figures for the system performance are presented based on a single "train" of the fuel preparation, combustor and gas turbine and one-half of the steam section.

DESCRIPTION OF SYSTEMS

Six systems are considered in this chapter. They are designated as system 1, system 2, system 3-2, system 4, system 5-1, and system 5-2 in accordance with the numbering scheme in the overall Biomass project. Each of the systems is described below.

System 1

System 1 is the base case arrangement as specified by Biomass Energy Corporation. Figure 1.1 shows the schematic diagram of the process. It is described in detail by Bauer et al (1983). This base case system includes some high temperature (1675 F) combustion flue gases extracted from a point just before the heat exchanger and blended with the gases exiting the boiler which is then fed to the fuel dryer.

System 2

System 2, shown in Figure 4.1, does not have the high temp-

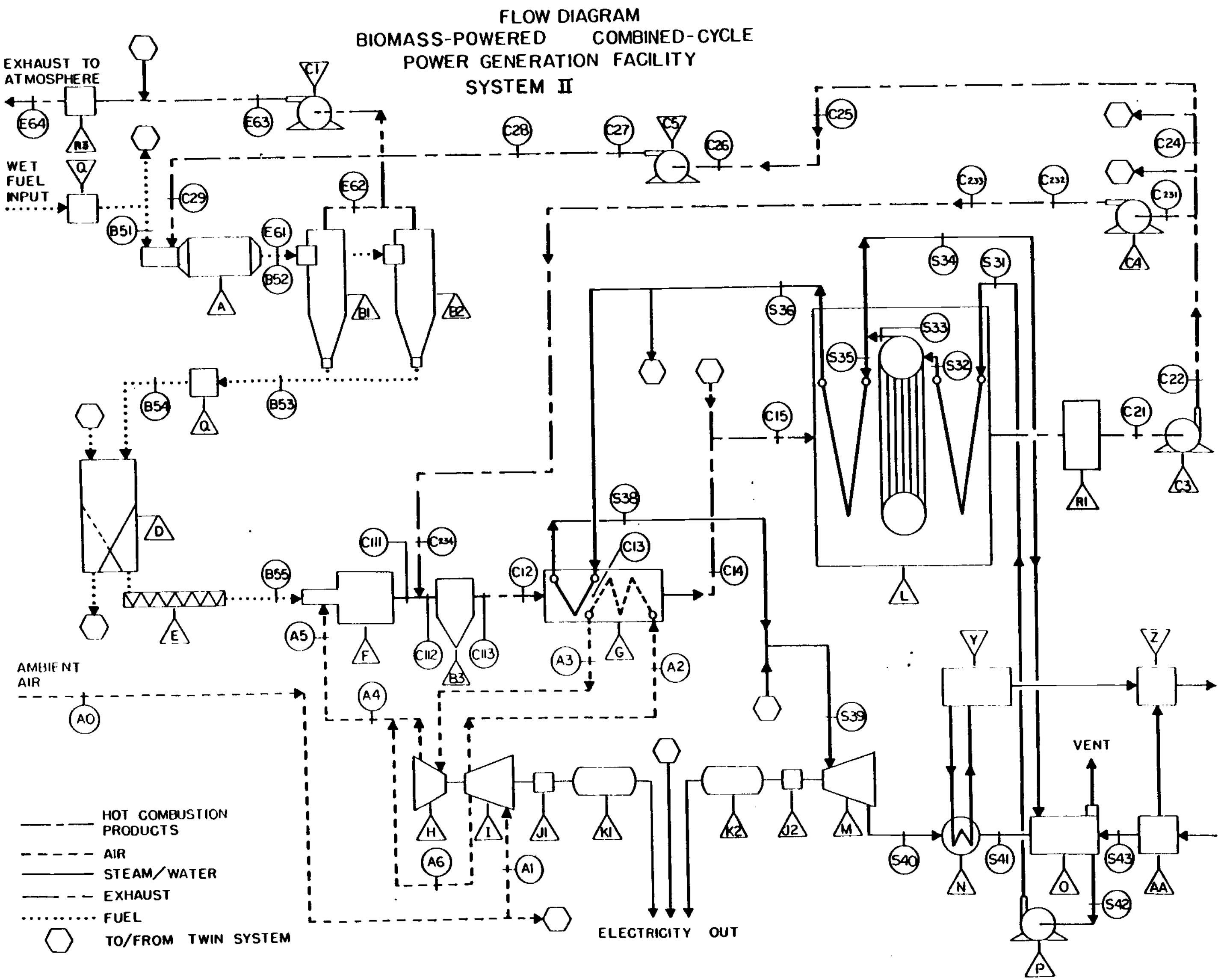


Figure 4.1 Schematic diagram of system 2. Refer to Table 1.1 for equipment nomenclature.

erature (1675 F) mixing stream c16, but is otherwise identical to the base case. In order to dry the same amount of fuel, the exit flue gas temperature from the boiler is adjusted from 350 F to 454 F. The air pollution equipment is also revised, because the elimination of stream c16 also eliminates the need for one pollution device. The differences between modeling the base case and system 2 with PROSIM follow.

	Base Case	System 2
Number of air pollution devices	2	1
Stream c16	allowed	not allowed
Boiler flue gas exit temperature	350 F	as required by dryer model

Although system 1 is termed the "base case", the remaining systems start with the configuration of system 2. They do not have stream c16, and most calculate the required temperature exiting the boiler, except in the cases where no dryers are used (systems 5-1 and 5-2).

System 3-2

System 3-2 adds steam injection to the configuration of system 2, as shown in Figure 4.2. Steam is extracted from the steam turbine and injected into the high pressure air exiting the compressor before the heat exchanger. The temperature and pressure of the steam is extracted at a point in the turbine that is similar to that of the air. This minimizes the irreversibility of the

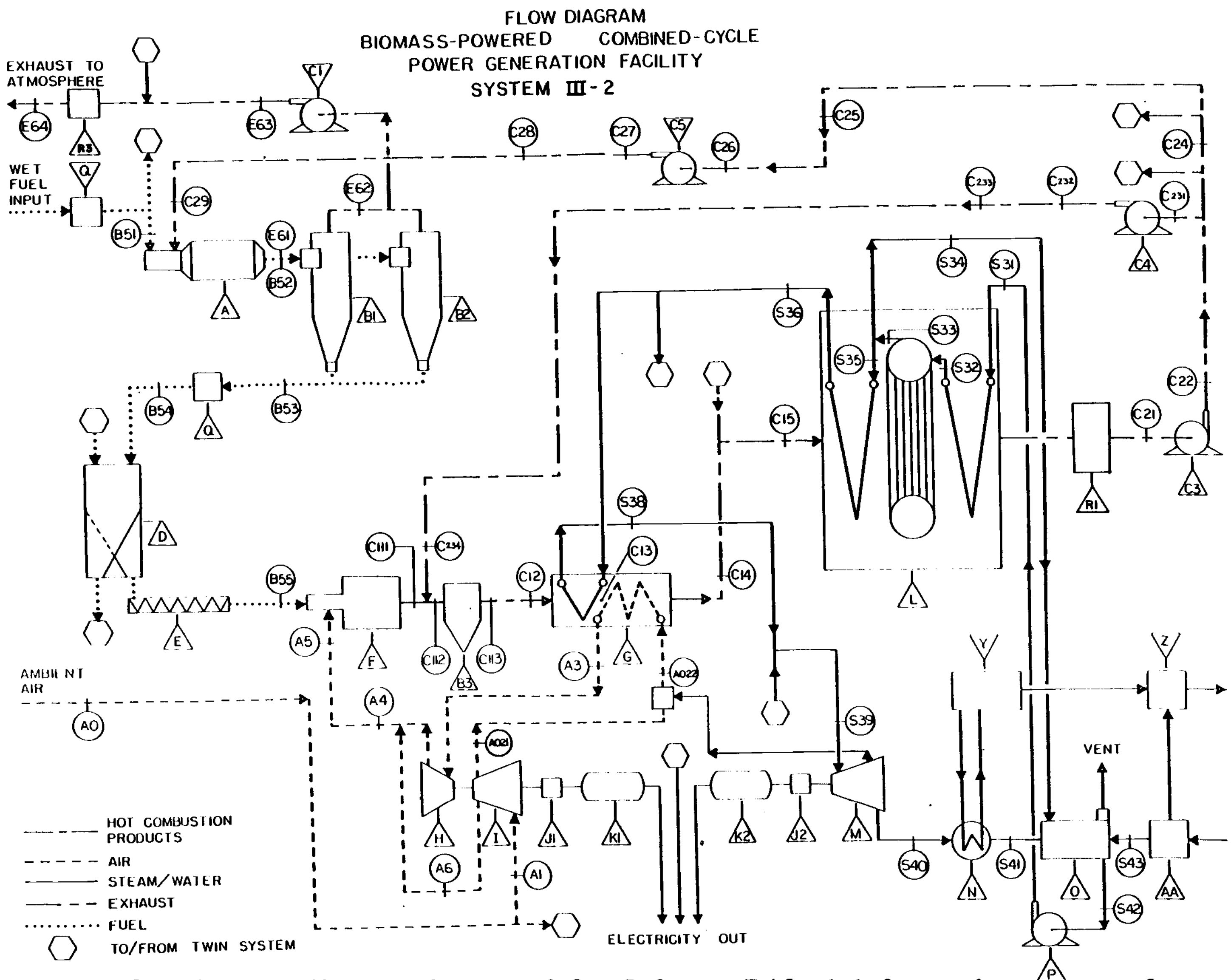


Figure 4.2 Schematic diagram of system 3-2. Refer to Table 1.1 for equipment nomenclature.

mixing. The differences in modeling system 3-2 and system 2 with PROSIM follow.

	System 2	System 3-2
Steam injection	none	5% of air mass flow rate
Steam turbine	one stage	two stage, with intermediate pressure 150 psia

A system identified as system 3-1 has previously been considered by the members of the Biomass project. It added steam injection to the air stream after the air-gas heat exchanger, before the gas turbine. Also, the steam was to be extracted from a point before the steam turbine. Computer runs with PROSIM indicated that system 3-1 had inferior efficiency, power output, and a large heat exchange area. The irreversibility of throttling the high pressure steam to the lower pressure of the air entering the gas turbine defeated the performance of this proposed system. This is the reason that system 3-2 was proposed.

System 4

System 4 differs from system 2 in that a trimburner is placed just before the gas turbine, as shown in Figure 4.3. Since the heat exchanger has a temperature limit, the exit air temperature is only 1450 F. However, the turbine is capable of a 1612 F inlet temperature, which may be achieved by burning a small amount of number 2 fuel oil with the 1450 F air. The difference in modeling the two with PROSIM follow.

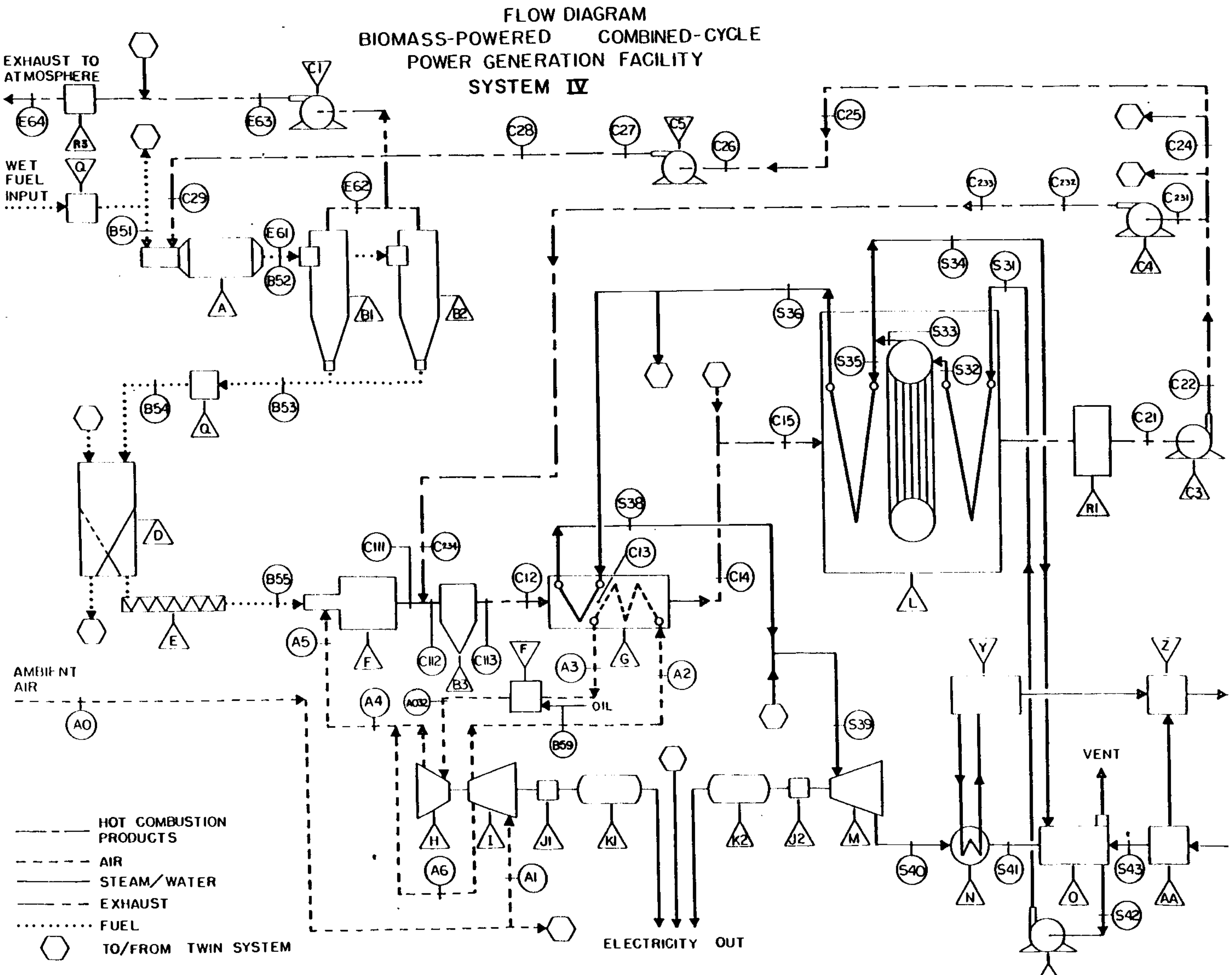


Figure 4.3 Schematic diagram of system 4. Refer to Table 1.1 for equipment nomenclature.

	System 2	System 4
Ta032	1450 F	1612 F

System 5-1

System 5-1 has the fuel preparation equipment removed. The removed equipment includes the dryer, hammermill, fans #1 and #5, and several pollution devices. Figure 4.4 shows the schematic of this system. The differences in modeling system 5-1 and system 2 follow.

	System 2	System 5-1
Dryer	used	not used
Hammermill power	used	not used
Bag house, cyclones	used	not used
Flue gas exit temp	190 F	350 F
Fuel moisture Wb52	12%	50%
Tc21	calculated by dryer	350 F

System 5-2

System 5-2 has the trimburner of system 4 without the fuel preparation equipment as in system 5-1.

RESULTS OF THE SIMULATIONS

Each power plant configuration was simulated using the computer program. The results are presented in pictorial form (Figures 4.5 through 4.11) and in Appendix E, which lists the final calculated values and the comparisons of each to system 2.

Figure 4.5 shows four of the efficiency calculations for each system. EFFHg is the plant gross efficiency based on the higher heating value (HHV) of the fuel. EFFHn is the plant net efficiency

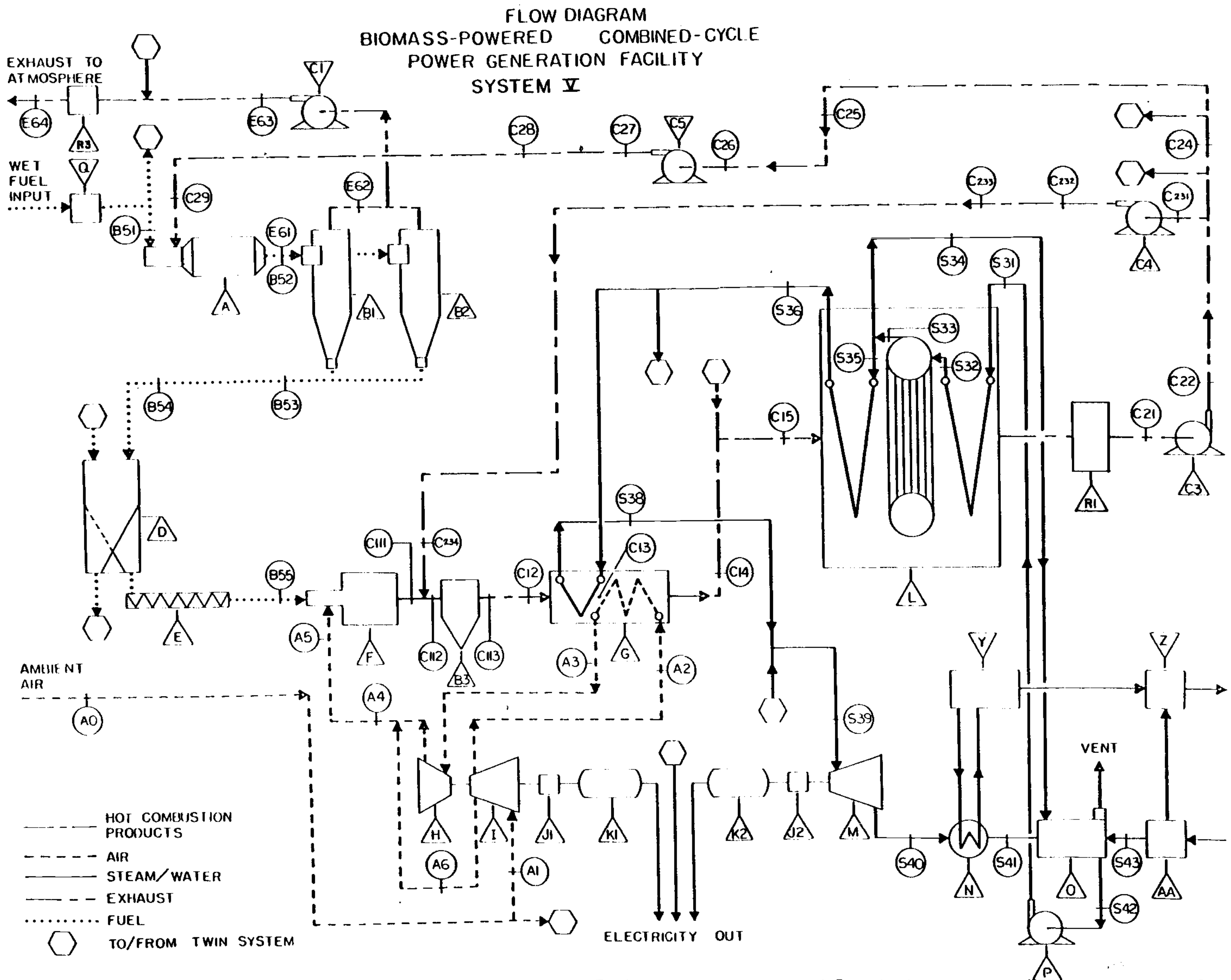


Figure 4.4 Schematic diagram of system 5-1. Refer to Table 1.1 for equipment nomenclature.

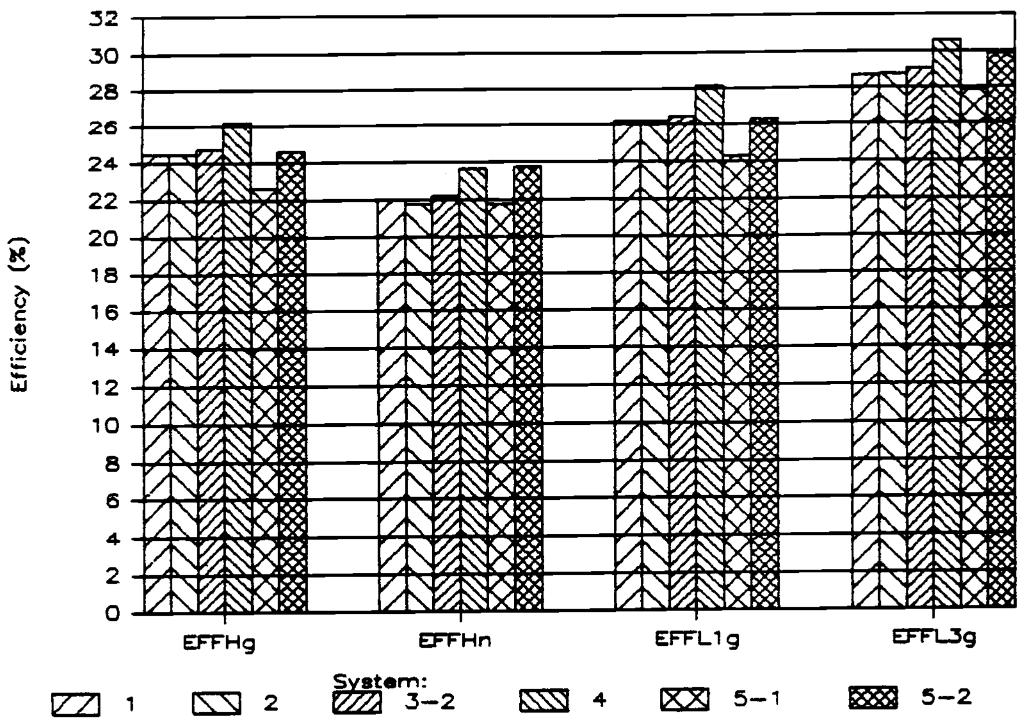


Figure 4.5 Gross and net efficiencies of the systems, based on higher heating value and lower heating value.

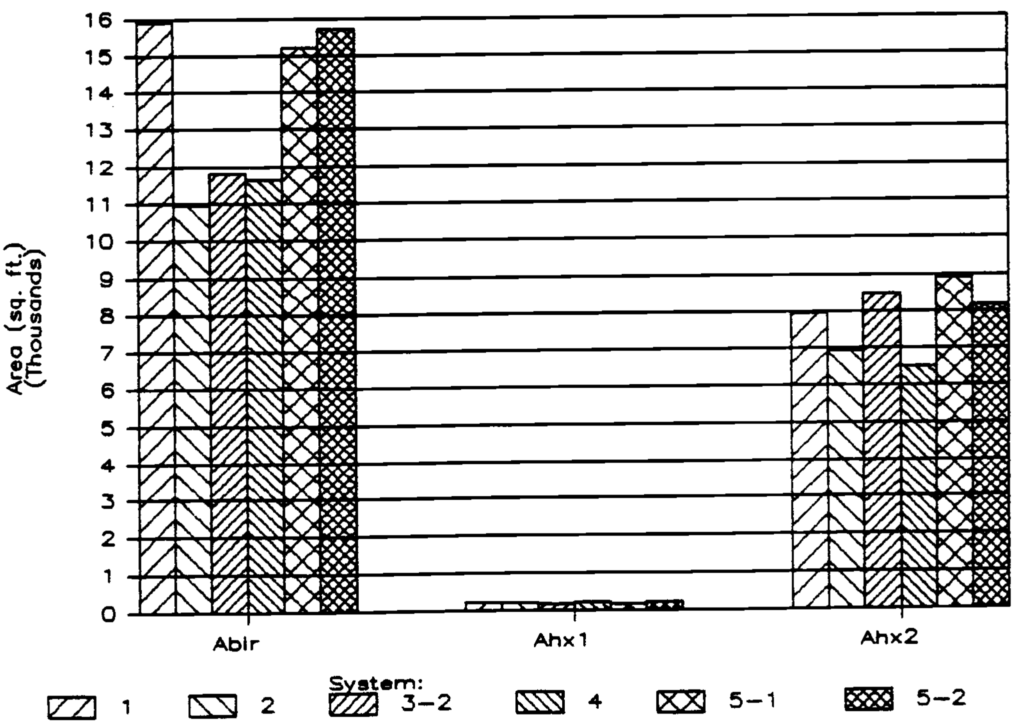


Figure 4.6 Areas of the boilers and heat exchangers of the systems.

based on HHV. The other two efficiencies are based on the lower heating values (LHV). LHV1 is equal to HHV reduced by the heat of vaporization of the water produced by combustion. LHV3 is equal to LHV1 reduced by the heat of vaporization of the water in the wood as received. EFFL1g is the plant gross efficiency based on LHV1; EFFL3g is the plant gross efficiency based on LHV3. Several efficiencies are reported so comparisons with other workers can be made, because different workers base efficiency on different criteria. EFFHn is the best indicator of the efficiency of the system.

Figure 4.6 shows the area required for the boiler (Ablr), steam-gas heat exchanger (Ahx1), and gas-air heat exchanger (Ahx2). The steam-gas heat exchanger is quite small compared to the other two, and does not vary much from system to system. The boiler size varies considerably. Systems 1, 5-1, and 5-2 require the largest boiler. The air-gas heat exchanger does not change as much as the boiler area, but is largest in the absence of a dryer (system 5-1). Direct comparisons of the systems with the trimburner (systems 4 and 5-2) cannot be made since there is more total fuel input to the plant.

Figure 4.7 shows the gross power output for each system, indicating the relative contribution of the gas turbine and steam turbine. Systems 1, 2, and 3-2 have closely the same total power output, but system 3 has a shift of power from the steam turbine to the gas turbine due to the steam injection. System 5-1 has a slightly lower gross power output. The two systems with a trim-

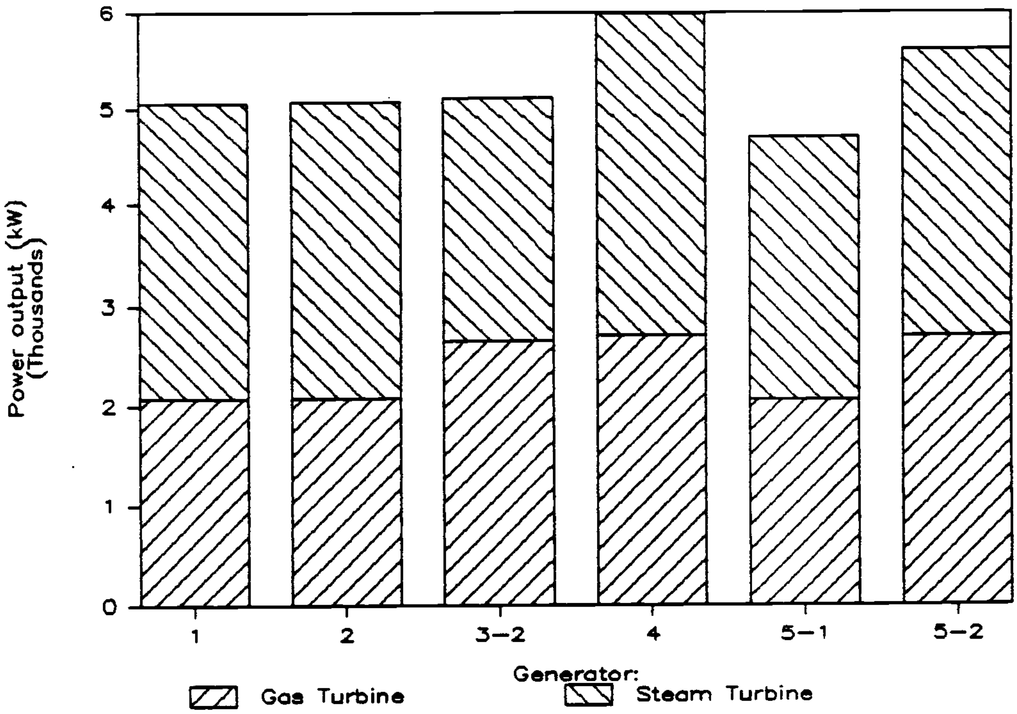


Figure 4.7 Gross power outputs of the systems.

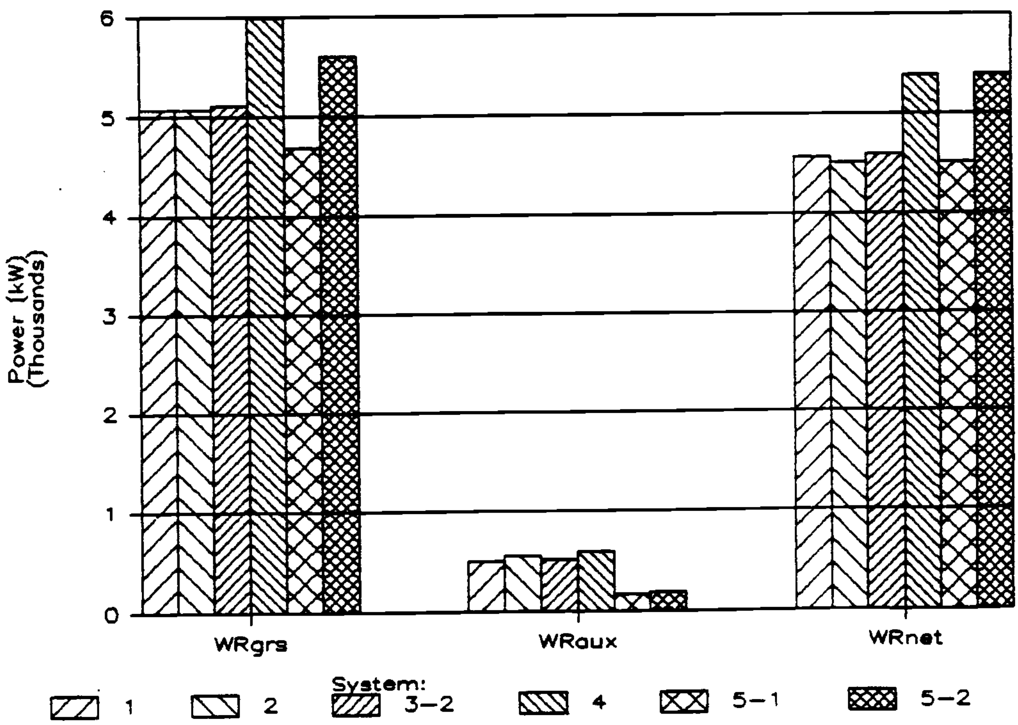


Figure 4.8 Gross, auxiliary, and net powers of the systems.

burner, systems 4 and 5-2, have substantially higher gross power outputs.

Figure 4.8 shows gross power output (WRgrs), auxiliary power output (WRaux), and net power output (WRnet). The net power is simply the gross power minus the auxiliary power. The systems without the dryers have less auxiliary power due to less recycling and no power requirements for the hammermill and dryer. The systems with trimburners have a higher gross and net power than the others, but the other systems have essentially the same net power outputs.

Figure 4.9 shows the irreversibility of the mixing streams. The two mixers are the cooling stream just after the combustor and the heating stream before the dryer. System 1 has a much higher irreversibility of mixing than the others. System 4 also has high mixing, due to the high temperature in the combustor.

Figure 4.10 shows the fan power requirement. There are four possible fans used in this group of systems, they are numbered #1, #3, #4 and #5. Figure 1.1 shows the location of each, shown as C1, C3, C4, and C5. The fans represent a large portion of the auxiliary power requirement of the plant.

Figure 4.11 shows three of the mass flow rates in the plant. MRc12 is the flue gas flow through the heat exchangers and boiler. MRc231 is the recycle flow that cools the combustor flue gas. MRs31 is the water flow rate through the boiler. These indicate which system uses the most recycling. The amount of recycling is dependent on the combustion temperature, since the temperature inlet to the

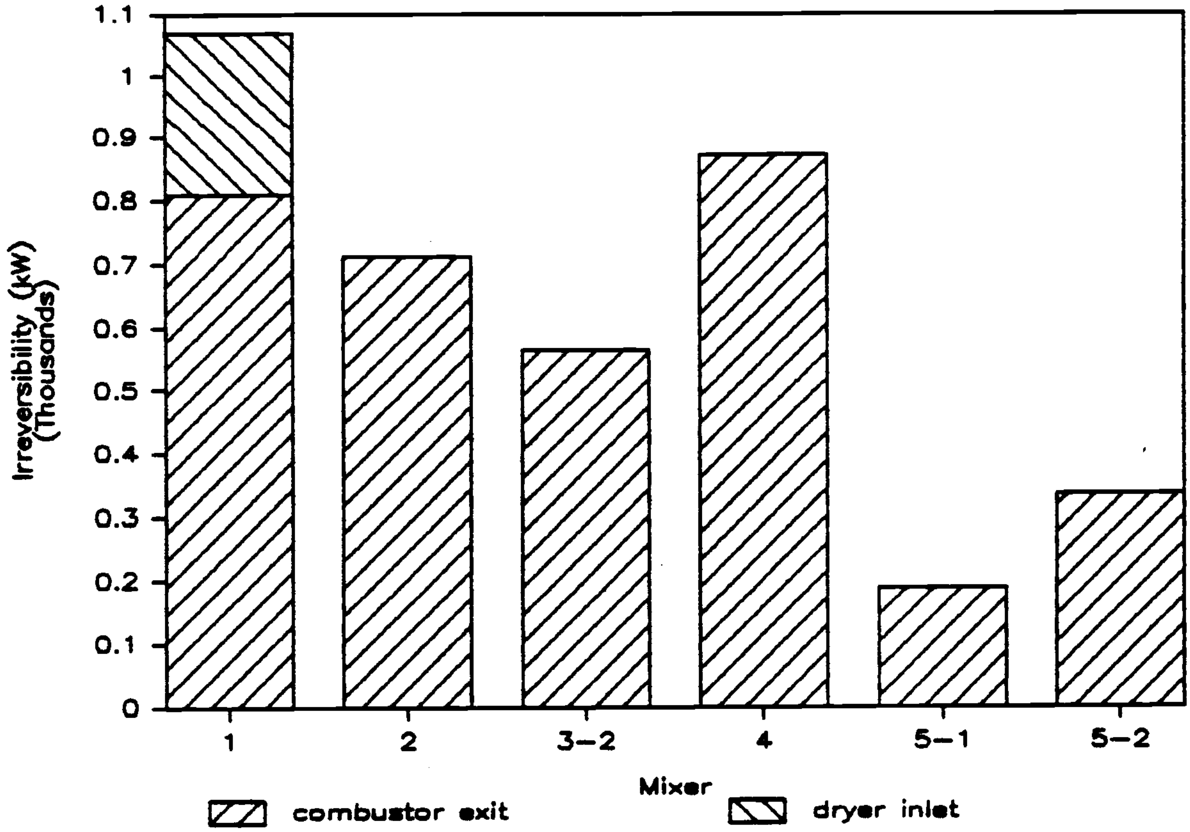


Figure 4.9 Irreversibilities of mixing of the systems.

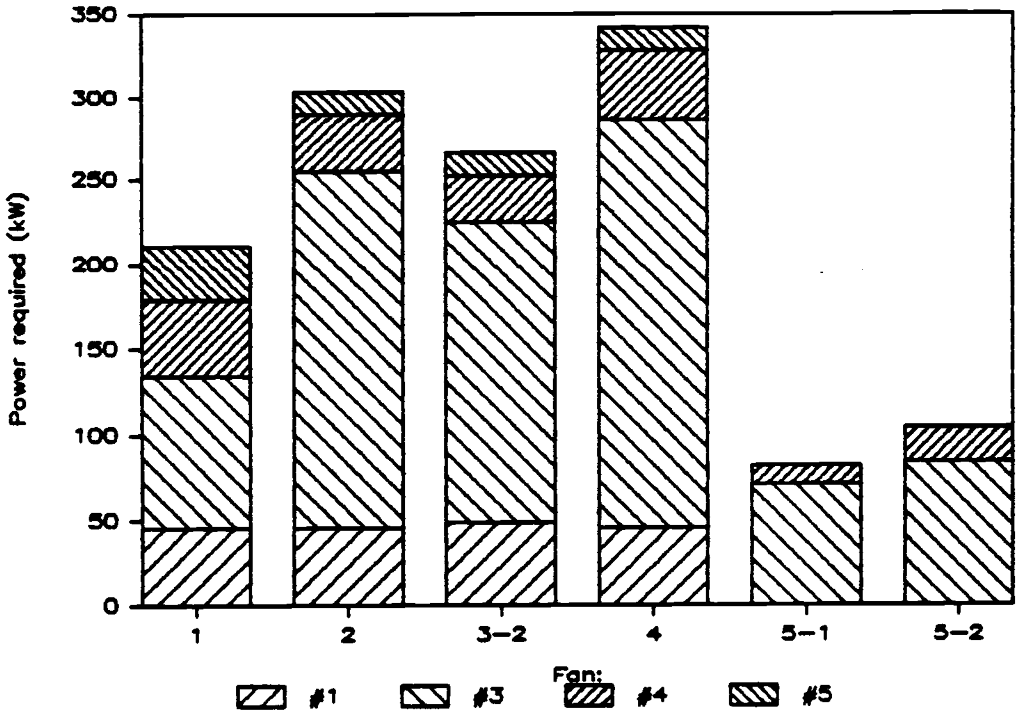


Figure 4.10 Powers required for the fans of the systems.

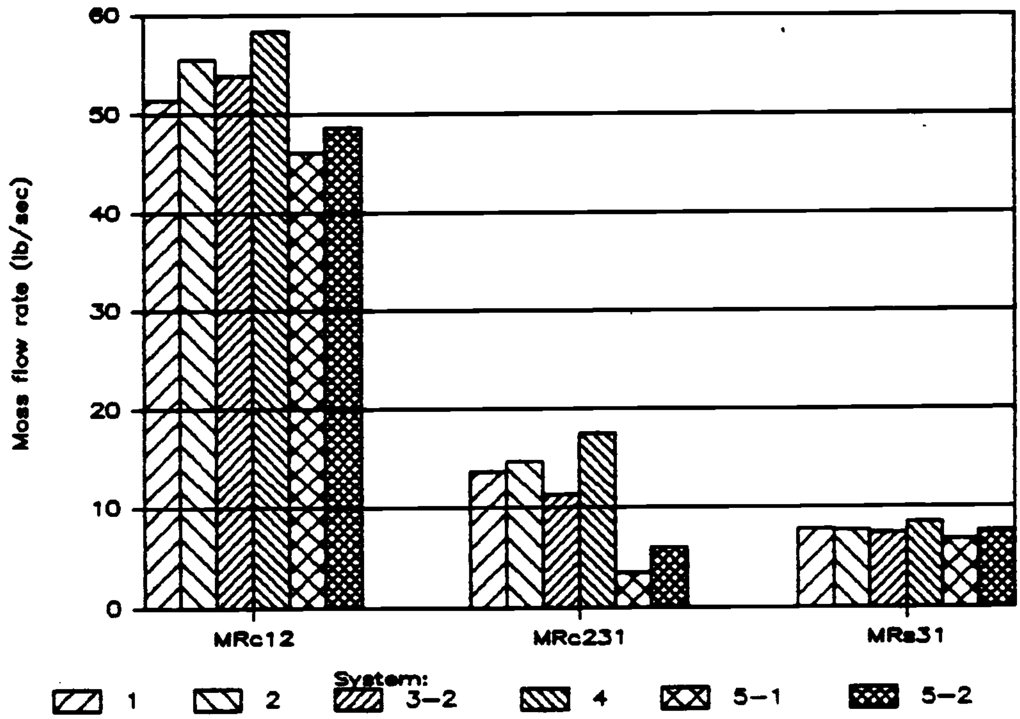


Figure 4.11 Flue gas, recycle, and steam flow rates of the systems.

heat exchanger is fixed at 1675 F. The systems with no drying have a lower combustion temperature due to the water in the fuel, and have less recycle flow. System 4 has a substantially greater recycle flow due to its higher combustion temperature, due to the higher temperature air into the combustor. The flow rate from the combustor is fairly constant for all the systems; it is MRc12 minus MRc231. The two systems with the trimburner have a higher steam flow rate, due to the higher total fuel into the plant.

Figures 4.12 and 4.13 show the irreversibilities of a few selected components in the plant. IRRblr is the irreversibility of the boiler, IRRcmb for the combustor, IRRhx1 for the steam-gas heat exchanger, and IRRhx2 for the air-gas heat exchanger. Note that the combustor irreversibility is a higher order of magnitude than the other components; it is therefore on a separate figure. Of particular significance is the higher irreversibility in the combustor when wet fuel is used. But a look at Appendix C shows that this is offset by the lack of a dryer in those systems, and the lower irreversibility of mixing due to the lower temperature out of the combustor.

COMPARISON OF SYSTEMS

System 1 (Base Case) versus System 2

Both the base case and system 2 have similar gross and net efficiencies, and gross and net power output. The base case has a very slightly higher net efficiency and net power. Both have the

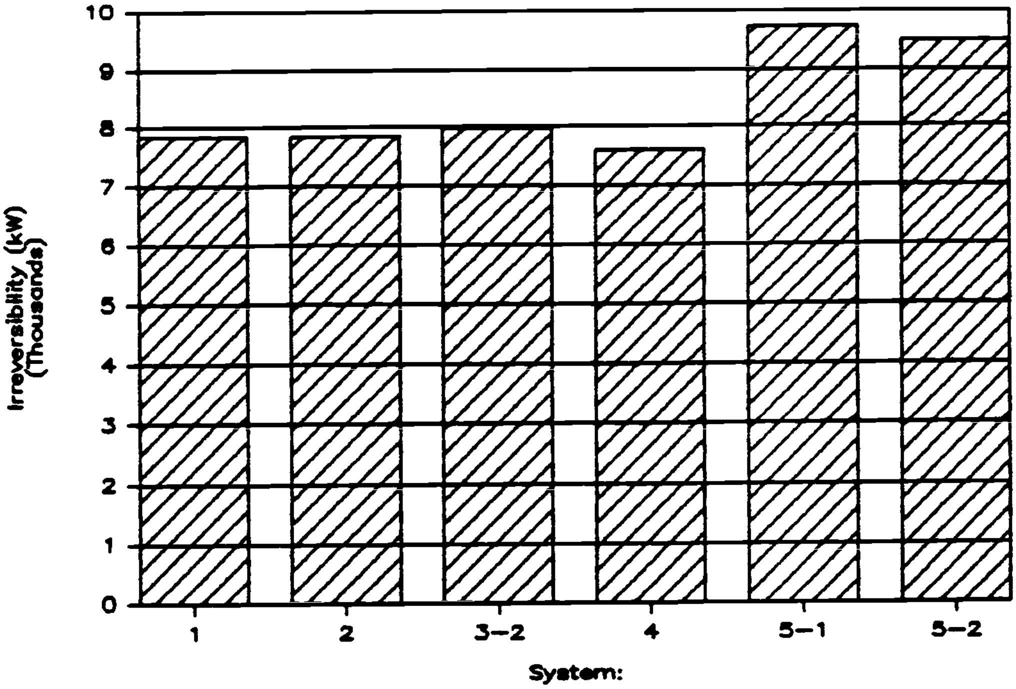


Figure 4.12 Irreversibilities of the combustors of the systems.

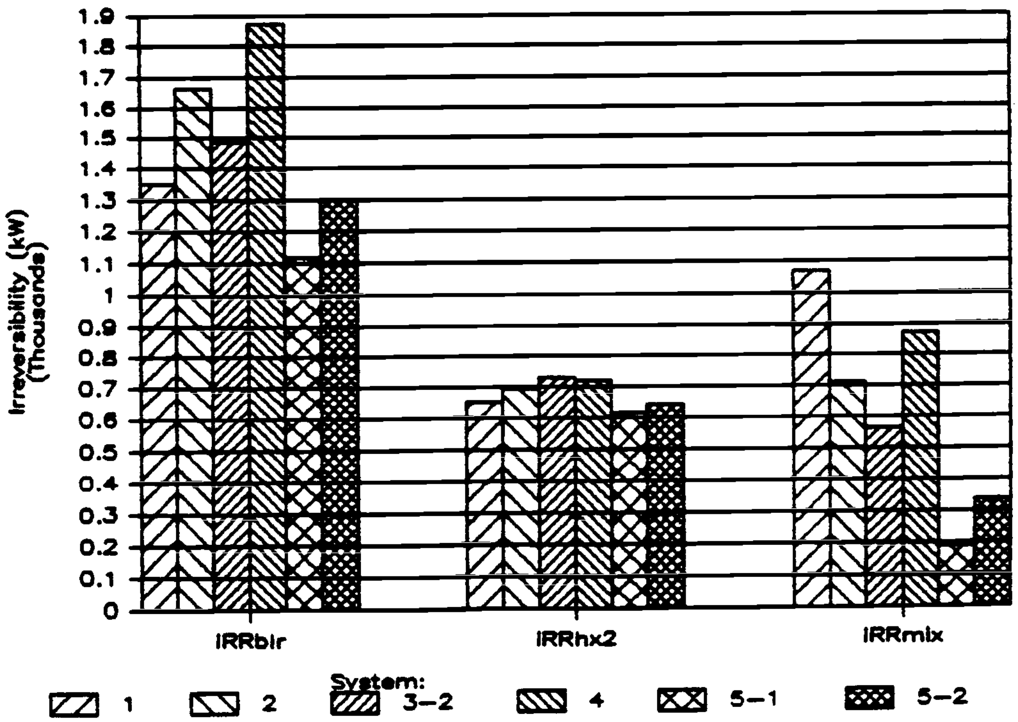


Figure 4.13 Irreversibilities of the boilers and heat exchangers of the systems.

same size dryer, combustor, gas turbine system, steam turbine, and condenser. However, the base case requires a 45% larger boiler and a 14% larger gas-air heat exchanger. The total additional heat exchange area is about 6000 square feet larger for the base case than for system 2. It required 60 kW (80 HP) more power for the fans than did the base case. But the relative costs of the additional heat exchange area versus the larger fans show that system 2 is a more attractive alternative.

The reason behind this can be seen by comparing the irreversibilities of the two systems. The following tabulation shows some of the major irreversibility differences between the two systems (in kW).

Component	Base Case	System 2	(System 2) -(Base Case)
Boiler	1347	1664	317
Gas-air heat exchanger	651	695	44
Fan #3	33	77	44
Mixer after combustor	808	713	-86
Mixer before dryer	261	0	-261
Total, all components	14883	14913	30

The system 2 boiler has 320 kW more irreversibility because the average temperature difference between the steam and flue gas is higher. The smaller heat exchanger area comes at the expense of a greater availability loss. The air-gas heat exchanger has more irreversibility for the same reason. Fan #3 has a higher flow rate, so its irreversibility is higher for system 2. But the two mixing streams almost make up for these three major differences. The mixer exiting the combustor has less irreversibility because the cooling stream temperature is lower (it is modeled as the same as that exiting the boiler). Of course, the irreversibility in the mixer involving stream c16 occurs only in the base case. When all the irreversibilities are added, system 2 has only 30 kW more than the base case system. This is very small when compared with the 4500 kW output the plant produces. The higher temperarature exiting the combustor cannot be used to advantage because of the temperature requirements of the heat exchanger. The irreversibility of the mixing in stream c16 does not have any advantage. In fact, it is a disadvantage because system 1 requires a much bigger boiler (i.e. greater expense).

System 3-2 versus System 2

The system 3-2 gross and net efficiency and power output are slightly higher than for system 2. The flow rate of "air" is 5% greater due to the injected steam. The air-gas heat exchanger area required is 22% larger, due to the higher flow rate of air/steam which causes the greater amount of heat transfer. The boiler area required is also larger by 8% due to the lower inlet gas temperature and mass flow rate. This increase in heat exchange area is more crucial than the higher net efficiency that would offset it, although an economic analysis would probably be required to substantiate this. Figure 4.7 shows the shift of power from the steam turbine section of the system. System 3-2 requires a substantially larger makeup water flow rate, totaling about 18900 gallons per day.

System 4 versus System 2

To raise Ta032 from 1450 F to 1612 F requires 0.1 lb/s of fuel oil, causing the net efficiency to raise from 21.8% to 23.7%. Thus, the oil is actually boosting the power output by slightly raising the temperature. The power output is raised from 4500 kW to 5400 kW by using 1987 btu/s (2094 kW) of fuel oil. Thus, the efficiency of the oil itself is 42%, which is very high. The temperature of the air exiting the gas turbine and entering the combustor is raised by about 100 F. Higher recirculation (MRc231) is necessary because the combustor exit temperature is higher resulting from the higher combustor inlet (turbine exit) temperature. The heat exchanger size is reduced by 6%, but the boiler has a 6% increase in size. The increase in heat exchange area is not too large considering that the net power output is increased by 20%. Both the gas turbine section and the steam turbine section have an increased power output, as shown in Figure 4.7.

System 5-1 versus System 2

Bauer (1984) studied system 5-1 extensively and reported that based on efficiency and power output, drying the fuel to 12% moisture (wet basis) does not significantly improve the system. The net efficiency shown in Figure 4.5 is approximately the same for system 2 and system 5-1. The heat exchanger area is greater, but some preliminary economic analysis has indicated that the savings in fuel preparation equipment make up for the increased cost in the boiler and heat exchanger.

System 4 and System 5-1 Combined

The advantages of both systems are clear in Figure 4.5, which shows a net efficiency as high as system 4. The boiler area is larger than that of system 5-1 but the heat exchanger is smaller. This is because more fuel is input to the system but the extra fuel does not affect the air stream until after it passes through the gas-air heat exchanger. This is transfered to the boiler and steam turbine. Figure 4.10 shows that system 4 requires much more fan power than system 5-2, due to the higher combustor temperature. The wet fuel helps to keep the recycling flue gas to a minimum.

The attractiveness of system 5-2 suggested that more study be made on some parametric evaluations. PROSIM was run varying the air-fuel ratio and the temperature exiting the trimburner. A higher A/F will lower the temperature in the combustor, reducing the need for recirculation at MRc231. It will also reduce the fuel flow and total power, since in the model, the air input is held constant, while the biomass fuel input is calculated from the air-fuel ratio (A/F). Higher temperature air exiting the trimburner will require more fuel, but should improve the overall efficiency.

Figure 4.14 shows several normalized results as a function of

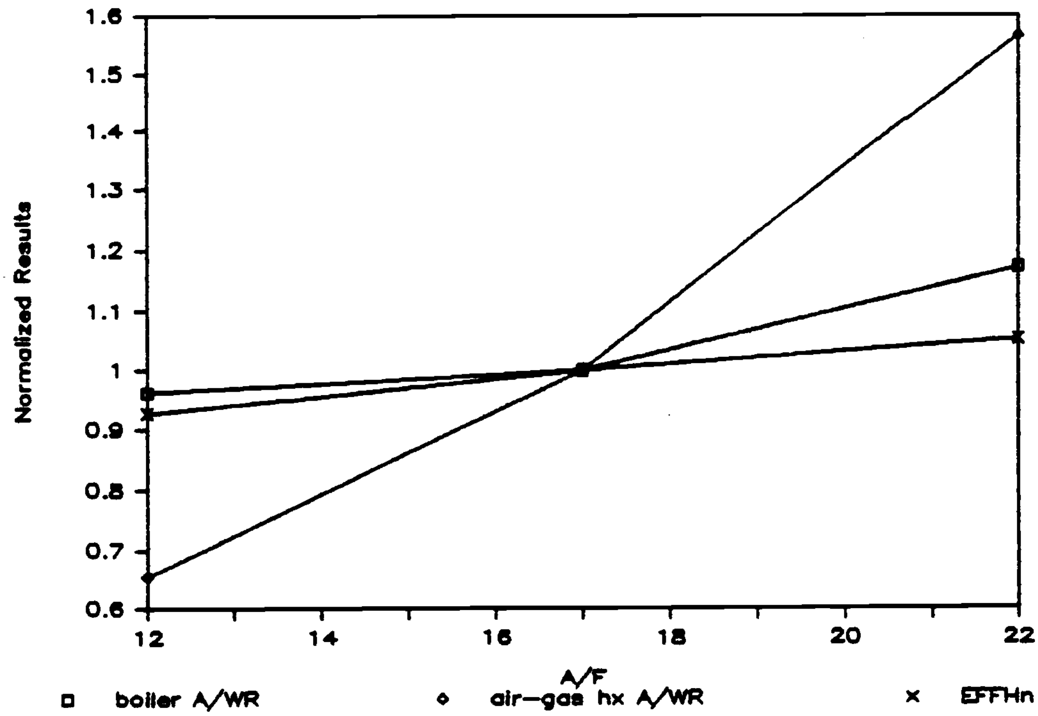


Figure 4.14 Areas per unit power and net efficiencies of system 5-2 as a function of air-fuel ratio, normalized at A/F = 17.

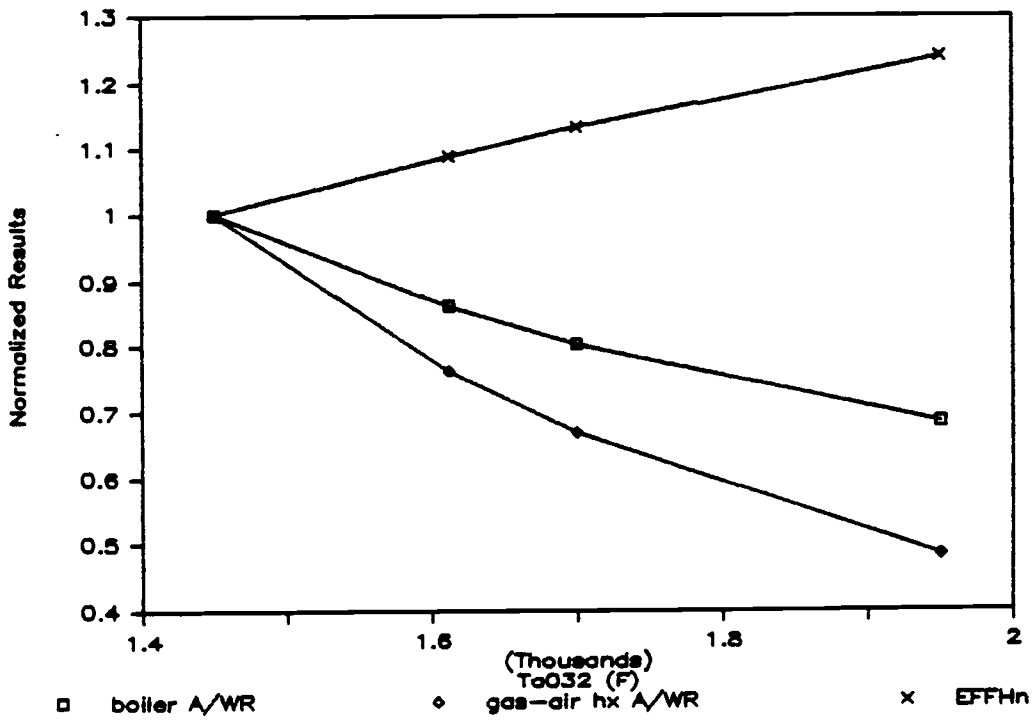


Figure 4.15 Areas per unit power and net efficiencies of system 5-2 as a function of the trimburner exit temperature, normalized at Ta032 = 1450 F.

air-fuel ratio. The area per unit power for the boiler and heat exchanger and the net efficiency are normalized with respect to their values at an A/F of 17 (the base case A/F). The efficiency rises a slight amount with A/F, but so do the areas per unit power. The effect on the boiler is not great; at A/F = 22 there is an increase of about 10% in square feet per kilowatt. This is due to the specified temperatures used in the simulations. However, the slope of the curve for the air-gas heat exchanger is much steeper. From A/F = 17 to A/F = 22 the area per unit net power increases by almost 60%, caused by a lower flow rate through it. The lower combustion temperature with high A/F means less recirculation is required at stream c231. (Recall that the primary air flow rate is held constant and therefore fuel flow rate decreases as A/F is increased.) Thus, the increase in area per unit net power is partly due to the decrease in power as A/F increases, due to the lower biomass fuel flow rate.

At A/F = 12, the air-gas heat exchanger area per unit power is reduced to 66% of the value at A/F = 17. The relative amount of fuel oil required decreases due to the greater amount of biomass fuel input to the system per unit flow rate of air. The following tabulation shows how some important parameters change with A/F, particularly that the biomass energy per unit fuel oil energy into the plant decreases with A/F.

A/F	Biomass energy / Fuel oil energy	Net Power,kW	Boiler Area,ft2	Heat exchanger Area,ft2
12	14.0	6200	19300	6800
17	9.9	5400	15700	8200
22	7.6	4500	15300	10700

Figure 4.15 shows normalized parameters as a function of the exit temperature of the trimburner (Ta032). Like Figure 4.14, the area per unit power for the boiler and heat exchanger and the net efficiency are normalized with respect to Ta032 = 1450 F. As expected, the efficiency increases, while the heat exchange areas decrease. Again, the relative costs must be compared to determine the optimum temperature. The base case system specified a gas turbine capable of 1612 F inlet temperature, which gives an appreciably better efficiency at a small cost increase (for the trimburner). However, the costs of gas turbines capable of higher inlet temperatures need to be weighed with the cost savings in the heat exchange area. The following tabulation demonstrates the amount of fuel oil required to maintain this high temperature.

Ta032 (F)	Fuel energy / Total energy	Net Power,kW	Boiler Area,ft2	Heat exchanger Area,ft2
1450	0.0%	4500	15200	8900
1612	9.2%	5400	15700	8200
1700	13.6%	5900	16000	7800
1950	23.9%	7400	17000	7000

CONCLUSIONS

The performance of various configurations of a biomass power plant was studied using a computer model developed to assist in these studies. Five system variations of the base case configuration were considered. Some parametric evaluations were studied for one of the better systems.

System 1, the base case, is not as attractive as system 2, which does not bleed part of the hot combustor exit gases to the dryer, because for the same net power and efficiency it uses a larger boiler. The irreversibility of mixing the very hot flue gas with the cooler flue gases going into the dryer is severe enough to require a much larger boiler to overcome it. System 3-2, which injects steam from the steam turbine into the gas turbine, does not have appreciably better results than system 2. System 4, which has a trimburner (and dryer), has a much higher power output and efficiency, but a large amount of flue gas recycle flow is necessary due to the high combustion temperature. System 5-1, which is the same as system 2 without the dryer and other fuel preparation equipment, has about the same efficiency as system 2. Its heat exchange area is larger, but it does not require a dryer. Fewer pieces of equipment are always desired for simplicity.

One system that appeared promising was system 5-2, which did not dry the biomass fuel and utilized a trimburner to boost the gas turbine inlet temperature. System 5-2 has the advantages of system 4 and system 5-1. By burning wet fuel the combustion temperature was lower, yet a sufficiently high turbine inlet temperature is realized by using the trimburner.

In general, the evaluations have shown that since high temperatures exiting the combustor cannot be utilized using the proposed air-gas heat exchanger in this plant (as it is limited to an inlet flue gas temperature of 1675 F), systems that have a lower combustion temperature (as long as it is above 1675 F) are more attractive. These systems have less flue gas recycle flow, and thus less irreversibility of mixing. It is more economical to have the irreversibility occur in the heat exchangers, since it results in a smaller area for the same amount of heat transfer due to the greater temperature difference between the two fluids in the heat exchangers.

The efficiencies of the various systems did not vary enough to determine which system was best, showing that purely thermodynamic comparisons are not enough. Detailed economic studies, which were not in the scope of this thesis, need to be done to determine the most attractive system considered here.

BIBLIOGRAPHY

Bauer, T., S. Fox, S. Brynjolfsson, G. Reistad, and D. Bushnell, "Combined Cycle Biomass Fueled Power Plant", Proceedings of the ASME Wood, Pulp and Forest Industries Technical Committee Conference, Oct 22, 1983, Corvallis, OR.

Bauer, T. L., Investigation of Combustor and Fuel Preparation Requirements for a Combined-Cycle Wood Fired Power Plant, M.S. Thesis, Oregon State University, Corvallis, 1984.

Benedek, P. (editor), Steady-state Flow-sheeting of Chemical Plants, 1980. Elsevier Scientific Pub. Co., Amsterdam.

Brynjolfsson, Sigurdur, "Investigation of Heat Exchangers for Application in Indirect Fired, Combined Cycle Power Plants Fired with Wood", M.S. project report, Department of Mechanical Engineering, Oregon State University, Corvallis, 1983.

Chen, C., and L. B. Evans, "Computer Programs for Chemical Engineers: 1978 -- Part I", Chemical Engineering, Vol. 85, No. 13, pp 145-154, June 5, 1978.

Chen, C., and L. B. Evans, "More Computer Programs for Chemical Engineers", Chemical Engineering, Vol. 86, No. 11, pp 167-168, May 21, 1979.

Cybernet Services, SYNTHA II Powerplant Design and Surveillance, Control Data Corporation, 1978.

Fewell, M. E., and N. R. Grandjean, User's Manual for Computer Code SOLTES-1 (Simulator of Large Thermal Energy Systems), Sandia Laboratories, Albuquerque, 1980.

Fischer, S. K., and C. K. Rice, A Steady-State Computer Design Model for Air to Air Heat Pumps, ORNL/CON-80, Dec 1981.

Fox, S., "The Water Property Functions", internal report, Biomass project, Department of Mechanical Engineering, Oregon State University, Corvallis, 1984.

Kennedy, T.C., "A Comparison of PROSIM and SYNTHA II", internal report, Biomass Project, Department of Mechanical Engineering, Oregon State University, Corvallis, 1983.

Parsons, M. G., and T. L. LaGuardia, "A Preliminary Heat-Balance Computer Program Conforming to SNAME T&R Bulletin 3-11", Marine Technology, Vol. 17, No. 2, April 1980, pp 131-162.

Posner, J., J. Hill, S. Miller, E. Gottheil, M. Davis, 1-2-3 User's Manual, Lotus Development Corporation, Cambridge, MA, 1983.

Stoecker, W. F., "A Generalized Program for Steady-State System Simulation", ASHRAE Transactions, 1971 Part 1, Vol. 77, pp. 140-148.

Stoecker, W. F., Design of Thermal Systems, 2nd ed., 1980. McGraw-Hill, New York.

Van Wylen, Gordon J., and Richard E. Sonntag, Fundamentals of Classical Thermodynamics, 2nd Ed SI version, 1976. Wiley & Sons, New York.

Zuck, I. Z., Thermodynamic Properties of Water for Computer Simulation of Power Plants, NUREG/CR-2518 R7, 1981. NRC FIN A4065.

APPENDICES

APPENDIX A

Using Program FLOWDIAG

The FLOWDIAG Program

Flow Diagrams for the Biomass Plant

Appendix A

FLOWDIAG

Using program FLOWDIAG

Flow diagrams are used to help graphically display the flow of information among modules in a system. Each module has its own box, with the information streams into and out of the module shown as lines drawn into and out of the box, respectively. An entire process can be represented as a series of boxes showing the exact inputs and outputs for each module. Each box is labeled with the name of its module, and each information stream is labeled with its respective name.

FLOWDIAG draws each box and its corresponding information stream inputs and outputs. The program also lists in alphabetical order all the information stream names after all the boxes are drawn. It checks to see if more than one box calculates a variable or result. It also checks for general FLOWDIAG input data errors. This can be useful in debugging the input file.

FLOWDIAG requires an input file, which is created with an editor. The input file consists of "keywords" and "names" entered in a certain format. The following sections describe the format, keywords, and names.

FORMAT

There are two types of "words" in the input data file, keywords and names. Keywords must start in column 1; names must start in other than column 1. Otherwise, the input is free format, i.e. words are separated by space(s) and/or lines.

KEYWORDS

Keywords are specific words that perform a function or describe what type of names are to follow that keyword. The seven keywords are box, varin, varex, consin, calcin, calcex, and print. These keywords must start in column 1 to be recognized as keywords (otherwise, the program will interpret them as names). Keywords must only consist of lower case letters.

Order of the keywords, except for the "box" keyword, is not important, so for example "varex" can be before "varin", and vice versa. Not all keywords are required to draw a particular box. If, for example, a particular box doesn't have any constants, the "consin" keyword does not have to be specified for that box.

However, the "box" keyword is order important, as it signifies a new box and causes the previous box to be drawn.

Here is a list of each keyword.

box: signifies the start of a new box. This keyword is followed by the box name. The "box" keyword and its corresponding box name must be at the beginning of the input file.

varin: the variables into the module, which are drawn at the top towards the middle.

varex: the variables out of the module, which are drawn at the bottom towards the middle.

consin: the constants into the module, which are drawn at the top towards the left.

calcin: the calculated results into the module, which are drawn at the top towards the right.

calcex: the calculated results out of the module, which are drawn at the bottom towards the right.

print: the results calculated in the module that are printed by that module. These are not passed to other modules. They are drawn at the right side of the box.

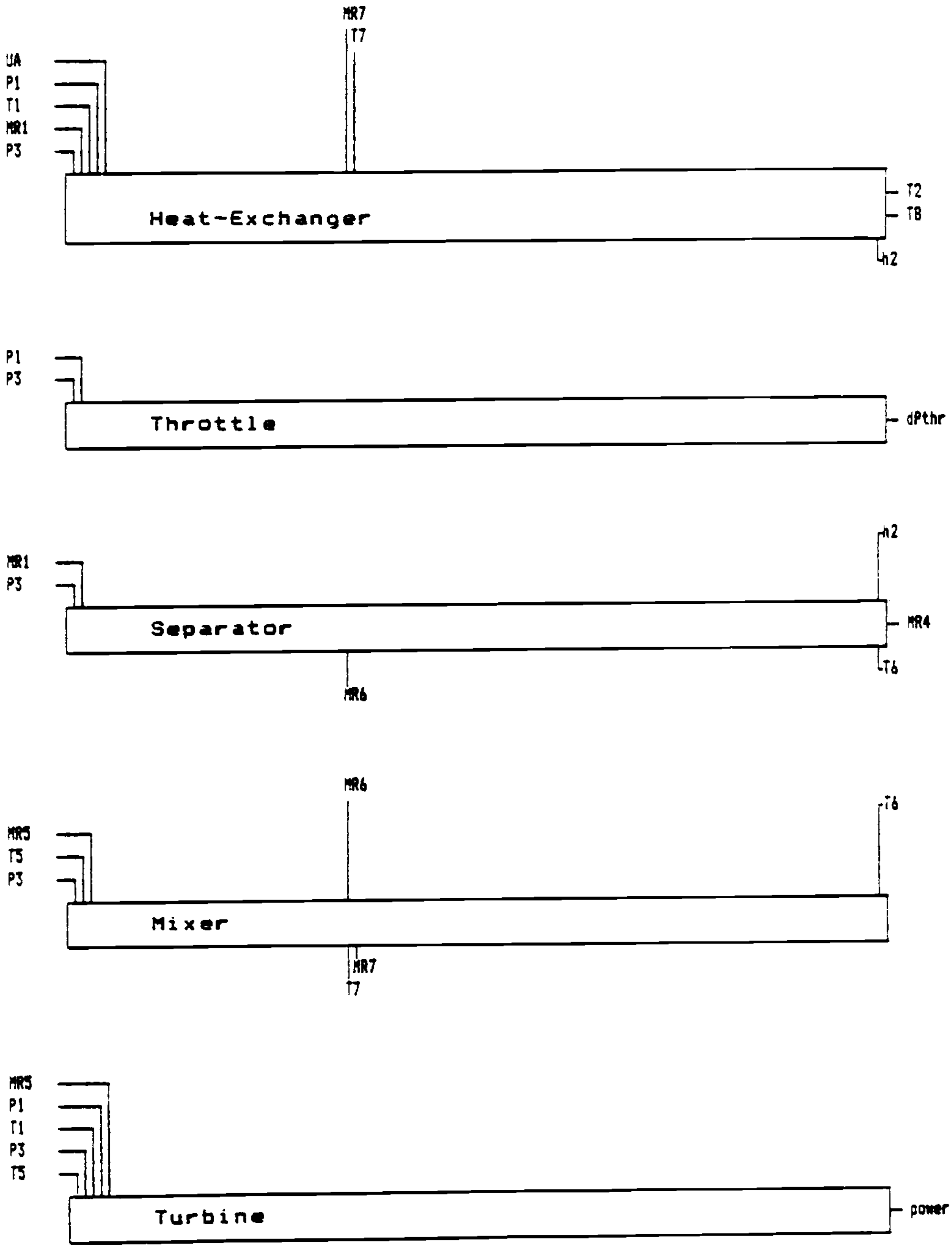
NAMES

Names may be up to 6 characters in length (to match the length of FORTRAN identifiers, which they represent); box names may be up to 40 characters long. These names may contain any character on the keyboard except the space bar. If multiple words are desired in a name, use a separating character such as "_", "-", ".", (for example: STEAM-HEAT-EXCHR).

SAMPLE INPUT FILE (and corresponding output)

```
box
   Heat-Exchanger
consin UA P1 T1 MR1 P3
varin MR7 T7
print T2 T8
calcex h2
box
   Throttle
consin P1 P3
print dPthr
box
   Separator
consin MR1 P3
calcin h2
varex MR6
calcex T6
print MR4
```

```
box
   Mixer
consin MR5 T5 P3
varin MR6
calcin T6
varex MR7 T7
box
   Turbine
consin MR5 P1 T1
       P3 T5
print power
```


MR7
T7
UA
P1
T1
MR1
P3
Heat-Exchanger
T2
T8
h2
P1
P3
Throttle
dPthr
h2
MR1
P3
Separator
MR4
T6
MR6
MR6
T6
MR5
T5
P3
Mixer
MR7
T7
MR5
P1
T1
P3
T5
Turbine
power

```
program FLOWDIAG
  (input.output,input_file,box_print_file,summary_print_file);
{----------------------------------------------------------}
{ FLOWDIAG -- draws information flow diagrams.             }
{     Variables in & out, constants in, intermediate       }
{     calculations, and internal printed results are       }
{     connected to a module box by arrows.                 }
{     The various names for each type of arrow are placed }
{     into an alphabetical binary tree and printed.        }
{ written by S. Fox 84.02.01                               }
{----------------------------------------------------------}
const
  max_names = 33;     { max no. of names per type per box }
  alfa_size = 6;      { max size a name can be }
  box =    'box    ';
  blank = '       ';
  boxname_size = 40; { max size the box name can be }
  pagesize = 63;      { lines per page }
type
  alfa = packed array[1..alfa_size] of char;
  box_alfa = packed array[1..boxname_size] of char;
  arrowtypes = (consin,varin,calcin,varex,calcex,print);
  arrow_array = array[arrowtypes] of record
    no:  0..max_names;
    name: array[1..max_names] of alfa
  end;
  nameptr = ^namestorage;
  namestorage = record
    name: alfa;
    before,after: nameptr
  end;
  nameptrs = array[arrowtypes] of nameptr;
  total_array = array[arrowtypes] of integer;

procedure date(var s:string); external;
procedure time(var s:string); external;

procedure skipblanks(var infile: text);
{----- skips blanks until non-blank or eof or eoln found --}
var finished: boolean;
begin
  finished := FALSE;
  repeat
    if      eof(infile)    then finished := TRUE
    else if eoln(infile)   then finished := TRUE
    else if infile^ <> ' ' then finished := TRUE
    else get(infile); { get next char }
```

```
  until finished;
end; { skipblanks }

procedure getboxname(var infile:text;
                     var boxname:box_alfa;
                     var error: boolean);
{----- gets boxname from 'infile' -----}
{----- assumes no leading blanks -----}
const
  blank = '                                            ';
var
  index: 0..boxname_size;
  finished: boolean;
begin
  boxname := blank;
  while (boxname = blank) and (not eof(infile)) do begin
    skipblanks(infile);
    if eoln(infile) then begin
      readln(infile);
      if infile^ <> ' ' then begin
        error := TRUE;
        writeln('box name error--start cont. line w/blank')
      end
    end
    else begin
      index := 0;
      finished := FALSE;
      repeat
        if      eof(infile)   then finished := TRUE
        else if eoln(infile)  then finished := TRUE
        else if infile^ = ' ' then finished := TRUE
        else begin
          if index < boxname_size then begin
            index := index + 1;
            read(infile,boxname[index])
          end
          else begin
            write('Warning: box name too long, trunc. to: ');
            writeln(boxname);
            get(infile)
          end
        end
      until finished;
      readln(infile)
    end
  end;
end; { getboxname }
```

```
procedure getstr(var infile:text; var str: alfa);
{----- gets a word 'str' from 'infile' -----}
{----- assumes no leading blanks -----}
var
  index: 0..alfa_size;
  finished: boolean;
begin
  str := blank;
  index := 0;
  finished := FALSE;
  repeat
    if      eof(infile)   then finished := TRUE
    else if eoln(infile)  then finished := TRUE
    else if infile^ = ' ' then finished := TRUE
    else begin
      if index < alfa_size then begin
        index := index + 1;
        read(infile,str[index])
      end
      else begin
        writeln('Warning: word too long, trunc. to: ',str);
        get(infile)
      end
    end
  until finished
end; { getstr }

procedure parse_line(var infile:text; var arrowtype:arrowtypes;
   var current_box:boolean; var arrow:arrow_array;
   var error:boolean);
{----- parses a line and fills arrow array -----}
{----- assumes starting at beginning of line -----}
var
  keyword: alfa;
begin
  {----- for lines containing a keyword -----}
  if infile^ <> ' ' then begin
    getstr(infile,keyword);
    if keyword = box then current_box := FALSE
    else if keyword = 'consin' then arrowtype := consin
    else if keyword = 'varin ' then arrowtype := varin
    else if keyword = 'calcin' then arrowtype := calcin
    else if keyword = 'varex ' then arrowtype := varex
    else if keyword = 'calcex' then arrowtype := calcex
    else if keyword = 'print ' then arrowtype := print
    else begin
```

```
      writeln('keyword ''',keyword,''' not in my vocabulary!');
      error := TRUE
    end
  end;
  skipblanks(infile);
  if current_box and (not error) then begin
    with arrow[arrowtype] do
      while (not eof(infile)) and (not eoln(infile)) and
          (no < max_names) and current_box do begin
        no := no + 1;
        getstr(infile,name[no]);
        skipblanks(infile);
        if (no = max_names) then
          writeln('Warning: ',max_names,' names read, the max')
      end;
    readln(infile)
  end
end; { parse_line }

procedure newpage(var outfile:text; arrow:arrow_array;
                  var current_line, current_page: integer);
{----- checks to see if room on current page -----}
var
  arrowtype: arrowtypes;
  lines_in_box: integer;
begin
  lines_in_box := 0;
  {----- total lines in current box -----}
  lines_in_box := lines_in_box + 5;
  if arrow[print].no = 0 then
    lines_in_box := lines_in_box + 1;
  for arrowtype := consin to print do
    lines_in_box := lines_in_box + arrow[arrowtype].no;
  {----- if a new page -----}
  if current_line + lines_in_box > pagesize then begin
    page(outfile);
    current_page := current_page + 1;
    writeln(outfile,chr(18),null:65,'page ',current_page:1);
    writeln(outfile,chr(15));
    current_line := 3 + lines_in_box
  end
  else
    current_line := current_line + lines_in_box;
    writeln(outfile);writeln(outfile);writeln(outfile)
end; { newpage }

procedure draw_box(var outfile:text;
```

```
          boxname:box_alfa; arrow:arrow_array;
          var current_line,current_page: integer);
{----- draws the information flow diagram boxes -----}
const
  vert_bar =  chr(179); horiz_bar = chr(196);
  top_T =     chr(194); bot_T =     chr(193);
  left_T =    chr(195); right_T =   chr(180);
  cross =     chr(197); top_left =  chr(218);
  top_right = chr(191); bot_left =  chr(192);
  bot_right = chr(217);
var
  col: 1..132; line: 1..max_names; i: 1..max_names;
begin
  newpage(outfile,arrow,current_line,current_page);
  {----- varin's -----}
  for i := 1 to arrow[varin].no do begin
    write(outfile,null:51);
    if i > 1 then
      for col := 1 to i-1 do write(outfile,vert_bar);
    writeln(outfile,arrow[varin].name[i])
  end;
  {----- calcin's -----}
  for i := 1 to arrow[calcin].no do begin
    write(outfile,null:51);
    for col := 1 to arrow[varin].no do write(outfile,vert_bar);
    write(outfile,null:67-arrow[varin].no-arrow[calcin].no);
    for col := 1 to i-1 do write(outfile,vert_bar);
    writeln(outfile,top_left,arrow[calcin].name[i])
  end;
  {----- consin's -----}
  for i := 1 to arrow[consin].no do begin
    write(outfile,null:9);
    write(outfile,arrow[consin].name[i],horiz_bar);
    for col := i to arrow[consin].no do
      write(outfile,horiz_bar);
    write(outfile,top_right);
    for col := 1 to i-1 do write(outfile,vert_bar);
    write(outfile,null:34-arrow[consin].no);
    for col := 1 to arrow[varin].no do write(outfile,vert_bar);
    write(outfile,null:67-arrow[varin].no-arrow[calcin].no);
    for col := 1 to arrow[calcin].no do write(outfile,vert_bar);
    writeln(outfile);
  end;
  {----- top of box -----}
  write(outfile,top_left:17);
  for col := 1 to arrow[consin].no do write(outfile,bot_T);
  for col := 1 to 34-arrow[consin].no do
```

```
  write(outfile,horiz_bar);
for col := 1 to arrow[varin].no do write(outfile,bot_T);
for col := 1 to 67-arrow[varin].no-arrow[calcin].no do
  write(outfile,horiz_bar);
for col := 1 to arrow[calcin].no do write(outfile,bot_T);
writeln(outfile,top_right);
{----- sides, inside of box, and local constants -----}
i := 1;
for line := 1 to arrow[print].no div 2 do begin
  writeln(outfile,vert_bar:17,left_T:102,horiz_bar,
    ' ',arrow[print].name[i]);
  i := i + 1
end;
write(outfile,vert_bar:17,null:10,chr(14),boxname,chr(20),
  null:11);
if arrow[print].no = 0 then
  writeln(outfile,vert_bar)
else begin
  writeln(outfile,left_T,horiz_bar,' ',
    arrow[print].name[i]);
  i := i + 1
end;
for line := i to arrow[print].no do
  writeln(outfile,vert_bar:17,left_T:102,horiz_bar,
    ' ',arrow[print].name[line]);
{----- bottom of box -----}
write(outfile,bot_left:17);
for col := 1 to 34 do write(outfile,horiz_bar);
for col := 1 to arrow[varex].no do write(outfile,top_T);
for col := 1 to 67-arrow[varex].no-arrow[calcex].no do
  write(outfile,horiz_bar);
for col := 1 to arrow[calcex].no do write(outfile,top_T);
writeln(outfile,bot_right);
{----- calcex's -----}
for i := 1 to arrow[calcex].no do begin
  write(outfile,null:51);
  for col := 1 to arrow[varex].no do
    write(outfile,vert_bar);
  write(outfile,null:67-arrow[varex].no-arrow[calcex].no);
  for col := 1 to arrow[calcex].no-i do write(outfile,vert_bar);
  write(outfile,bot_left);
  writeln(outfile,arrow[calcex].name[i])
end;
{----- varex's -----}
for i := 1 to arrow[varex].no do begin
  write(outfile,null:51);
  for col := 1 to arrow[varex].no-i do
```

```
      write(outfile,vert_bar);
    writeln(outfile,arrow[varex].name[i])
  end;
end; { draw_box }

function capitalized(str: alfa): alfa;
{----- returns capitalized 'str' -----}
var
  i: 1..alfa_size;
  temp_str: alfa;
  ordinal_value: 0..255;
begin
  temp_str := str;
  for i := 1 to alfa_size do begin
    ordinal_value := ord(temp_str[i]);
    if ordinal_value in [97..122] then
       temp_str[i] := chr(ordinal_value-32)
  end;
  capitalized := temp_str
end;

procedure addname(newname:alfa; arrowtype:arrowtypes;
                  var current: nameptr;
                  var total: integer);
{----- adds 'newname' to the link list -----}
{----- RECURSIVE procedure -----}
begin
  if current = nil then begin
    new(current);
    current^.name := newname;
    current^.before := nil;
    current^.after := nil;
    total := total + 1
  end
  else if capitalized(newname) < capitalized(current^.name) then
    addname(newname,arrowtype,current^.before,total)
  else if capitalized(newname) > capitalized(current^.name) then
    addname(newname,arrowtype,current^.after,total)
  else { newname = current^.name }
    if (arrowtype = varex) or (arrowtype = calcex) then
      writeln('Hey, ''',newname,''' exits more than one box')
end; { addname }

procedure build_binary_trees(arrow:arrow_array;
                    var rootptrs: nameptrs;
                    var total: total_array);
{----- builds up the binary tree structure to alphabetize
```

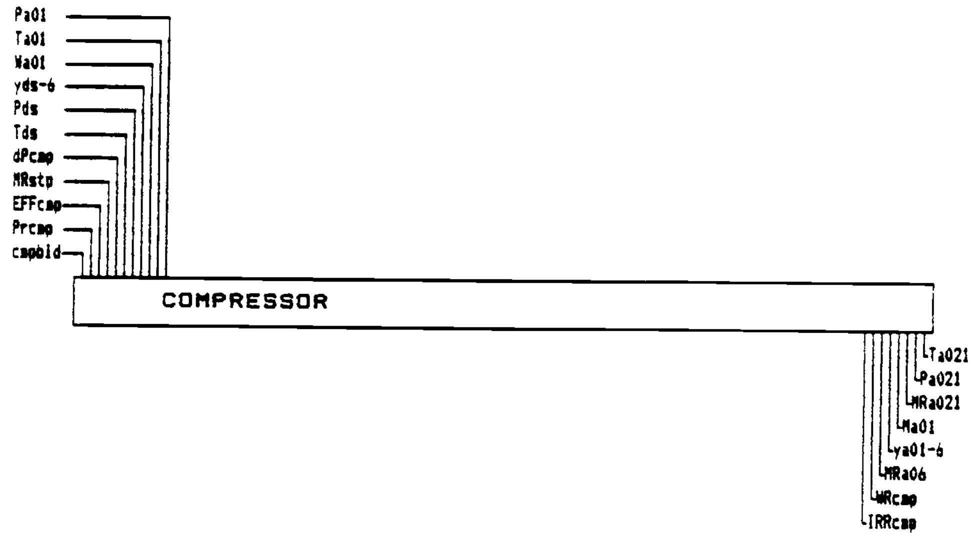
```
       the names in each category -----}
var i: 1..max_names;
    arrowtype: arrowtypes;
begin
  for arrowtype := consin to print do
    for i := 1 to arrow[arrowtype].no do
      addname(arrow[arrowtype].name[i],arrowtype,
              rootptrs[arrowtype],total[arrowtype]);
end; { build_binary_tree }

procedure print_tree(var f:text; current_node:nameptr);
{----- recursively visits each node of a binary tree -----}
begin
  if current_node <> nil then begin
    if current_node^.before <> nil then
      print_tree(f,current_node^.before);
    write(f,current_node^.name:10);
    if current_node^.after <> nil then
      print_tree(f,current_node^.after)
  end
end; { print_tree }

procedure print_binary_trees(var outfile:text;
                             rootptrs:nameptrs;
                             total:total_array);
{----- prints each binary tree to a file -----}
var arrowtype: arrowtypes;
    name: alfa;
begin
  for arrowtype := consin to print do begin
    write(outfile,chr(14),chr(15));
    case arrowtype of
      consin     : write(outfile,'Constants:');
      varin      : write(outfile,'Variables in:');
      varex      : write(outfile,'Variables out:');
      calcin     : write(outfile,'Intermediate calculations input:');
      calcex     : write(outfile,'Intermediate calculations output:');
      print      : write(outfile,'Printed results:')
    end;
    write(outfile,' (',total[arrowtype]:1,')');
    writeln(outfile,chr(18));
    print_tree(outfile,rootptrs[arrowtype]);
    writeln(outfile); writeln(outfile);
  end
end; { print_binary_tree }

procedure search_tree(name:alfa; current_node:nameptr; var found:boolean);
```

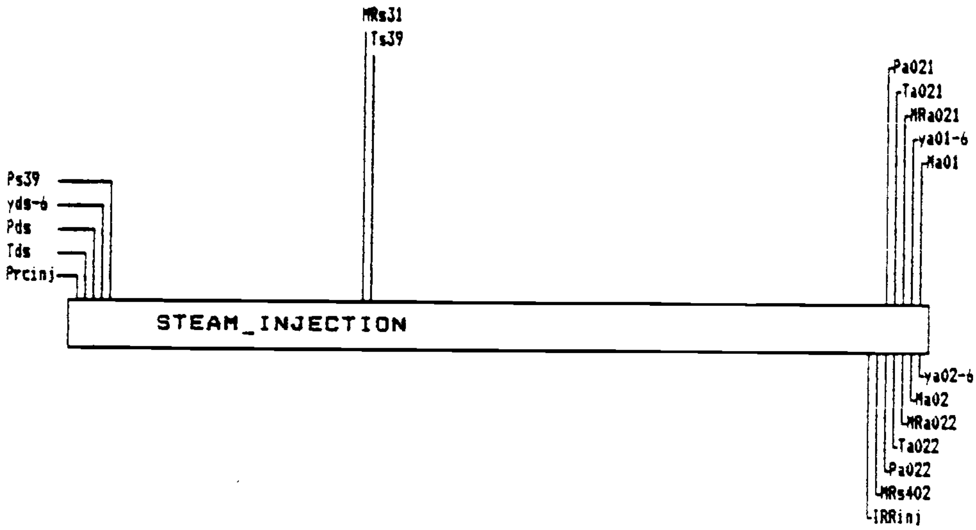
```
{----- searches binary tree for "name" -----}
begin
  if current_node <> nil then begin
    if current_node^.before <> nil then
      search_tree(name,current_node^.before,found);
    if name = current_node^.name then
      found := true;
    if current_node^.after <> nil then
      search_tree(name,current_node^.after,found)
  end
end; { search_tree }

procedure compare_tree(compare_root:nameptr; current_node:nameptr;
                       compare_type,current_type: alfa);
{----- compares current_node name with each element in tree starting at
       compare_root -----}
var
  found: boolean;
begin
  if current_node <> nil then begin
    if current_node^.before <> nil then
      compare_tree(compare_root,current_node^.before,compare_type,current_type);
    found := false;
    search_tree(current_node^.name,compare_root,found);
    if not found then
      writeln(current_type,' ''',current_node^.name,''' is not yet a ',
              compare_type);
    if current_node^.after <> nil then
      compare_tree(compare_root,current_node^.after,compare_type,current_type)
  end
end; { compare_tree }

{========== MAIN PROGRAM, FOLKS =============}
var
  arrowtype: arrowtypes;
  arrow: arrow_array;
  error: boolean;
  keyword: alfa;          { a recognized string }
  current_box: boolean;
  current_line: integer;
  current_page: integer;
  input_file: text;       { input data file }
  box_print_file: text;      { boxes printed here }
  summary_print_file: text;   { alphabetical list printed here }
  total: total_array;     { total no. of each arrowtype }
  boxname: box_alfa;
  index: 1..max_names;
```

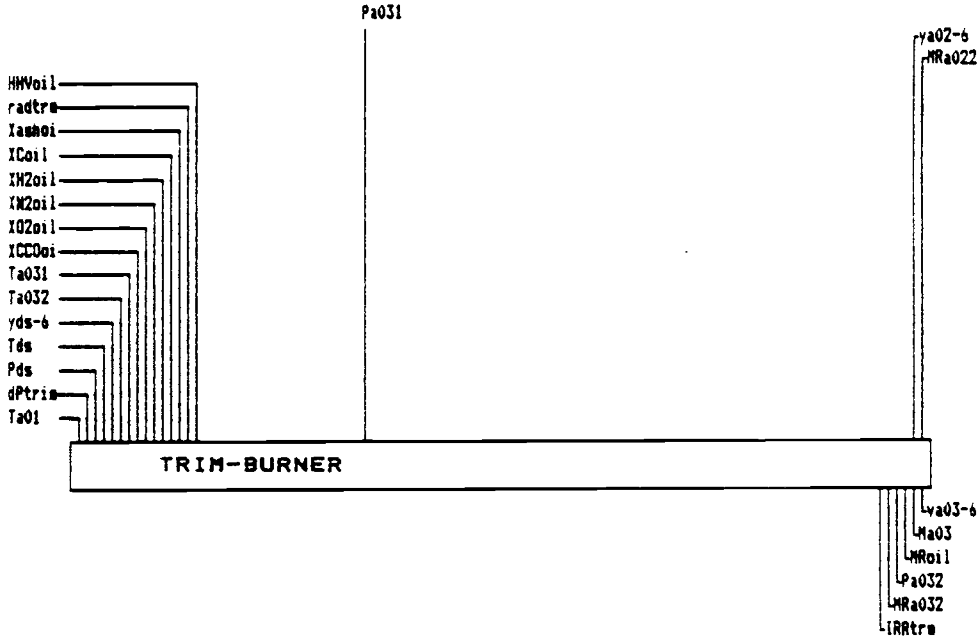
```
  rootptrs: nameptrs;   { points to roots of binary trees }
  today,now: string(8);
begin
  {----- initialize -----}
  reset(input_file);
  rewrite(box_print_file);
  error := FALSE;
  getstr(input_file,keyword);
  if keyword <> box then begin
    error := TRUE;
    writeln('box command missing.')
  end;
  for arrowtype := consin to print do begin
    total[arrowtype] := 0;
    rootptrs[arrowtype] := nil
  end;
  date(today); time(now);
  current_page := 1;
  writeln(box_print_file,chr(14),'   Flow Diagrams. ', chr(20),
          '    As of ', today, now:10,
          ' page ', current_page:1);
  current_line := 2;
  write(box_print_file,chr(27),'U',chr(1));  { unidirectional printing }
  write(box_print_file,chr(15));             { compressed printing }
  {----- process the input file -----}
  while (not eof(input_file)) and (not error) do begin
    {----- clear arrow array -----}
    for arrowtype := consin to print do begin
      arrow[arrowtype].no := 0;
      for index := 1 to max_names do
        arrow[arrowtype].name[index] := blank
    end;
    {----- get box info and draw the damn thing -----}
    getboxname(input_file,boxname,error);
    current_box := TRUE;
    while current_box and (not eof(input_file)) and (not error) do
      parse_line(input_file,arrowtype,current_box,arrow,error);
    {----- see if calcin's have been calcex's yet, then draw box -----}
    compare_tree(rootptrs[calcex],rootptrs[calcin],'calcex','calcin');
    draw_box(box_print_file,boxname,arrow,current_line,current_page);
    {----- build alphabetical link-list -----}
    build_binary_trees(arrow,rootptrs,total)
  end;
  {----- back to bidirectional printing -----}
  write(box_print_file,chr(27),'U',chr(0));
  write(box_print_file,chr(18));
  page(box_print_file);
```

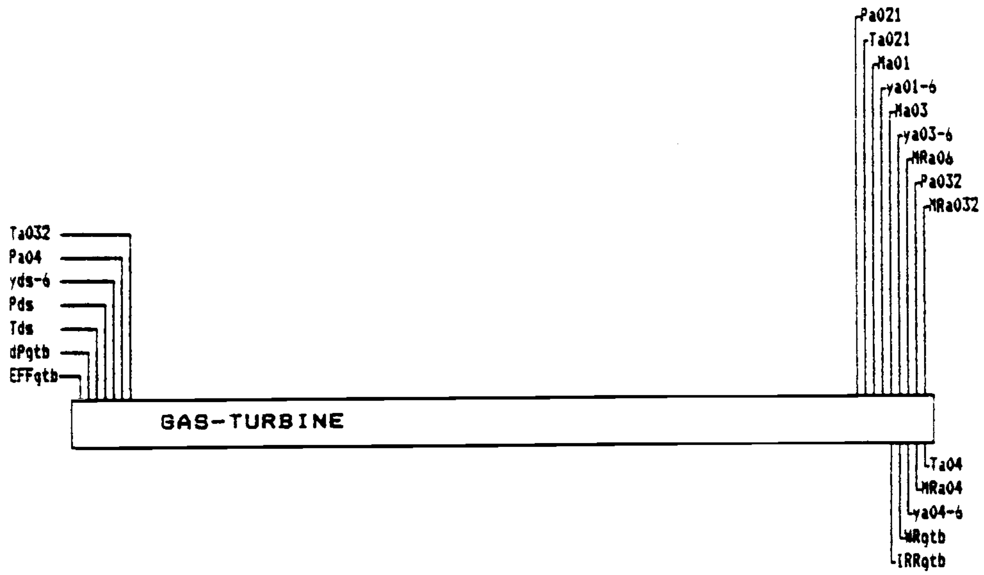
```
  close(box_print_file);
  {----- print the alphabetized lists -----}
  rewrite(summary_print_file);
  writeln(summary_print_file,chr(14),'Flow diagrams.',chr(15),
          'Name summary:',chr(20),chr(18),
          ' As of ', today, now:10);
  writeln(summary_print_file);
  print_binary_trees(summary_print_file,rootptrs,total);
  {----- check varin's: are they also varex's? -----}
  compare_tree(rootptrs[varex],rootptrs[varin],'varex ','varin ');
  {----- all done, folks -----}
  close(input_file);
  close(summary_print_file);
  writeln('End of FLOWDIAG.   ',chr(001))
end.
```

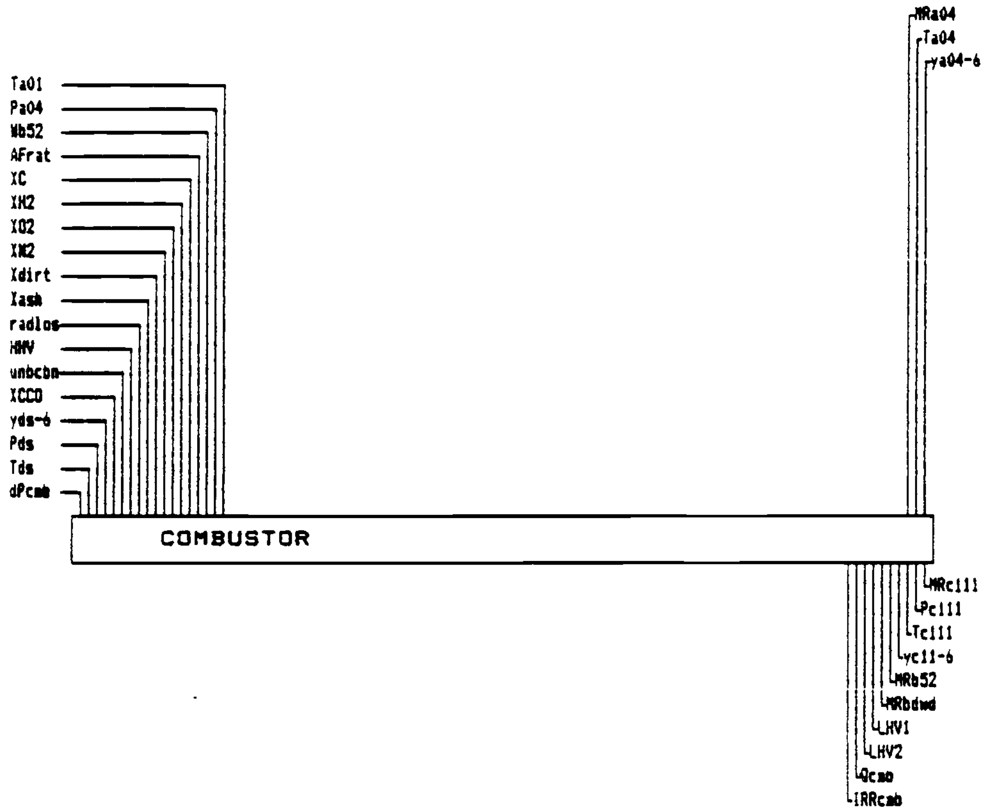
Flow Diagrams for the Biomass Plant

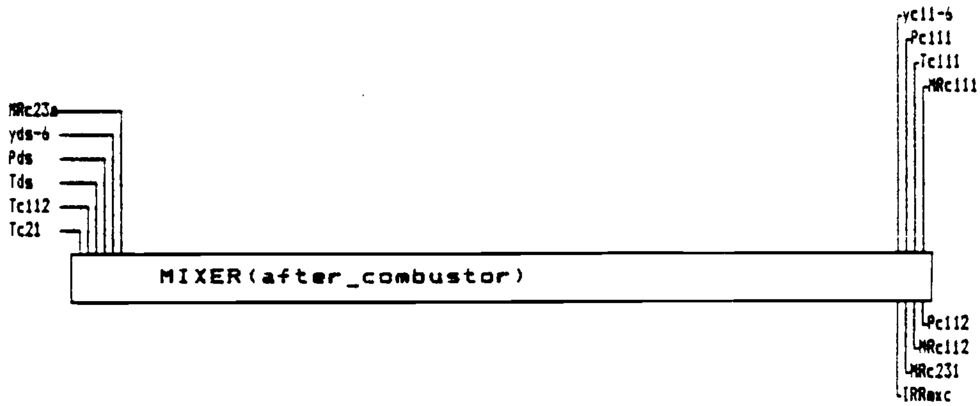


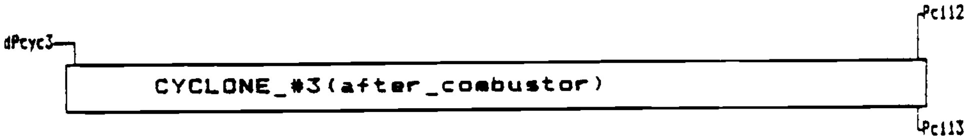


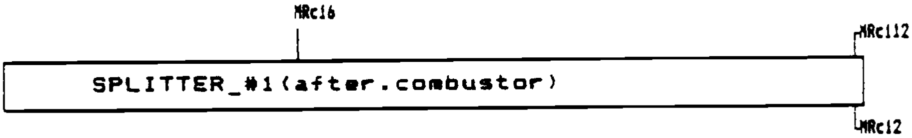

Pa031
ya02-6
MRa022
HHVoil
radtrm
Xashoi
XCoil
XH2oil
XN2oil
XO2oil
XCCOoi
Ta031
Ta032
yds-6
Tds
Pds
dPtrim
Ta01
TRIM-BURNER
ya03-6
Ma03
MRoil
Pa032
MRa032
IRRtrm

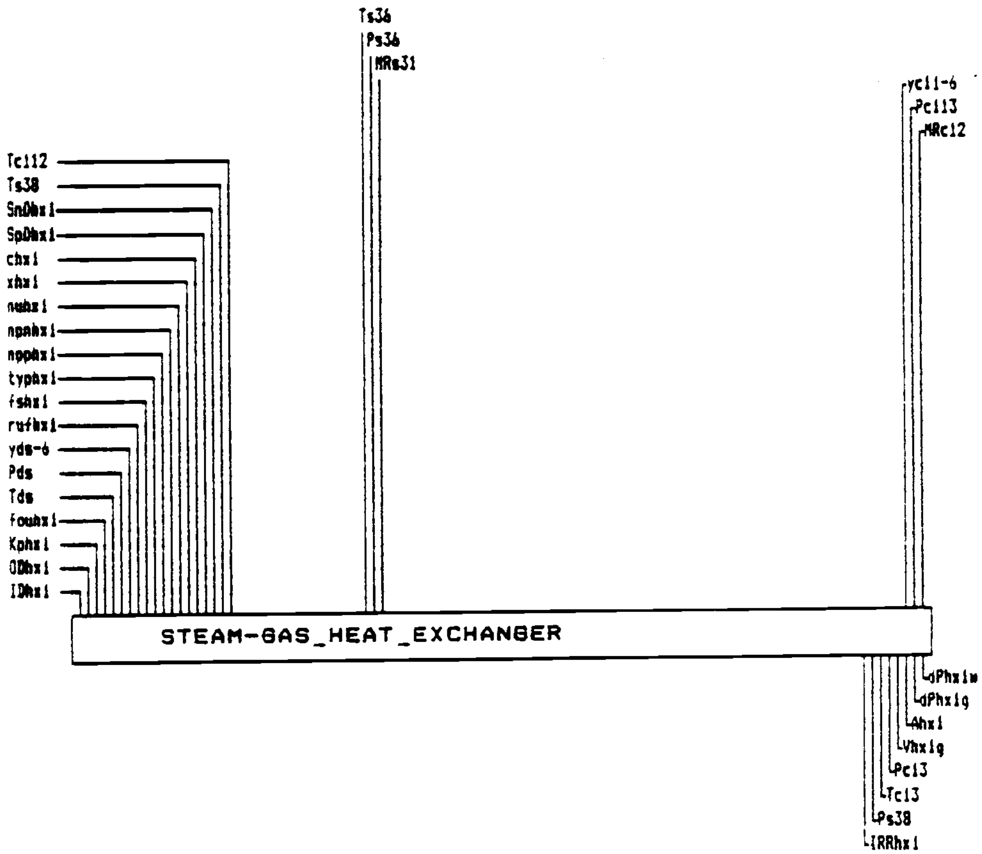

Pa021
Ta021
Ma01
ya01-6
Ma03
ya03-6
MRa06
Pa032
MRa032
Ta032
Pa04
yds-6
Pds
Tds
dPgtb
EFFgtb
GAS-TURBINE
Ta04
MRa04
ya04-6
WRgtb
IRRgtb

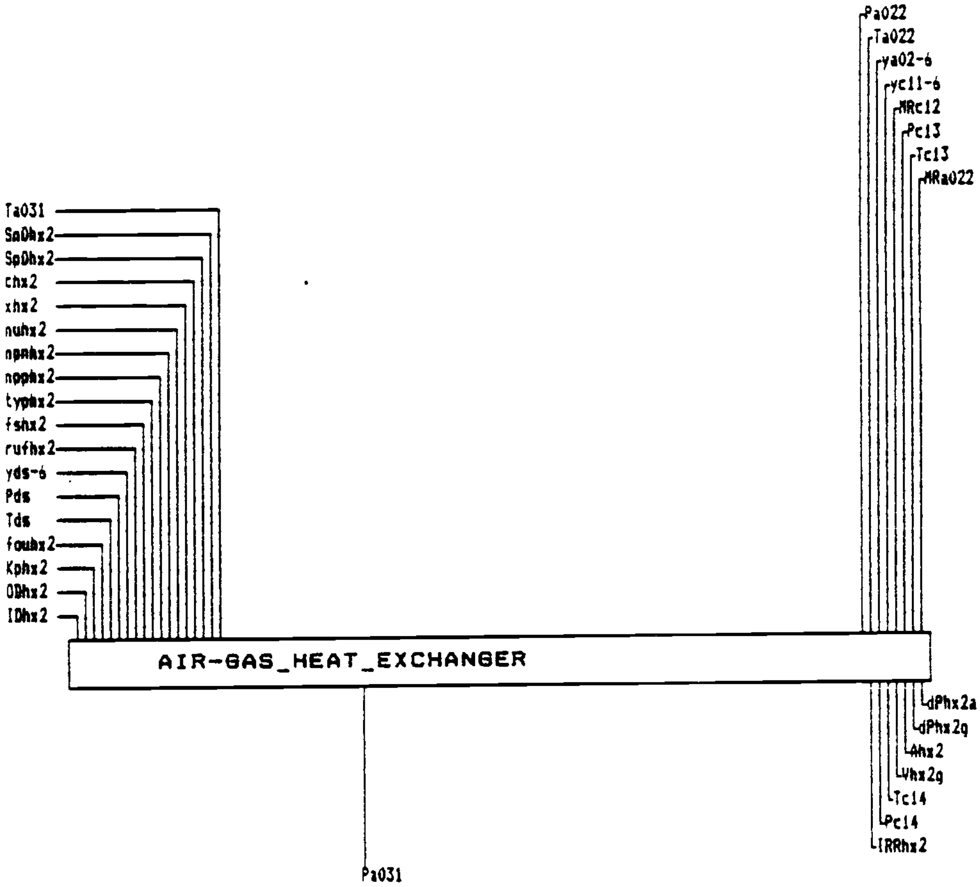

Ta01
Pa04
Wb52
AFrat
XC
XH2
XO2
XN2
Xdirt
Xash
radlos
HHV
unbcbn
XCCO
yds-6
Pds
Tds
dPcmb
MRa04
Ta04
ya04-6
COMBUSTOR
MRc111
Pc111
Tc111
yc11-6
MRb52
MRbdwd
LHV1
LHV2
Qcmb
IRRcmb

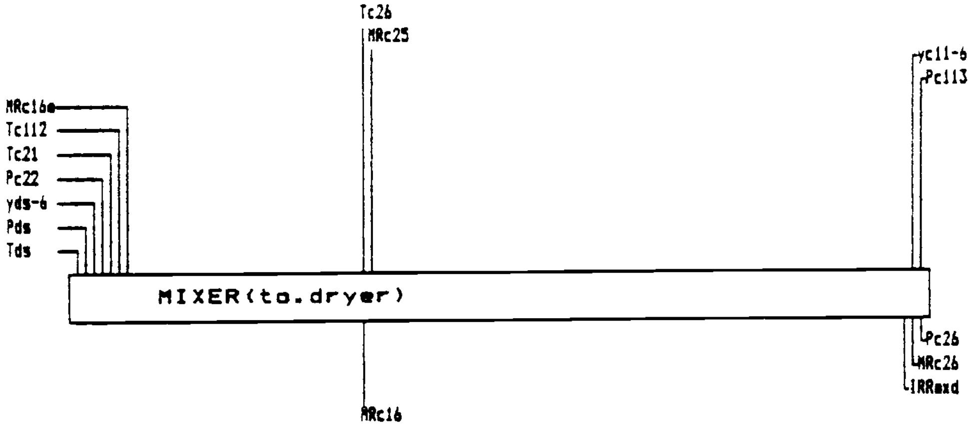

MRc23a
yds-6
Pds
Tds
Tc112
Tc21
yc11-6
Pc111
Tc111
MRc111
MIXER(after_combustor)
Pc112
MRc112
MRc231
IRRmxc


dPcyc3
Pc112
CYCLONE_#3(after_combustor)
Pc113

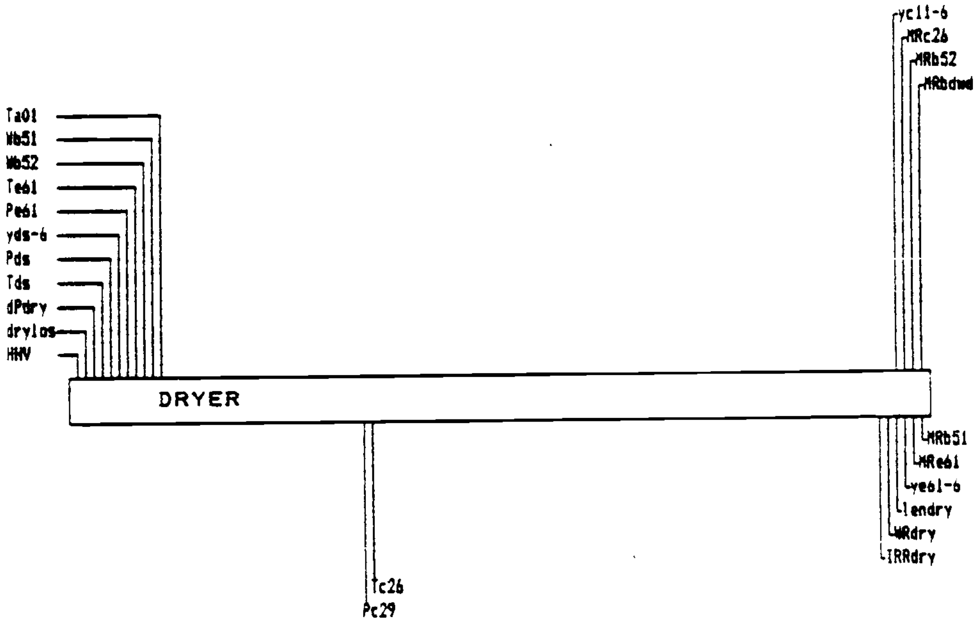

MRc16
MRc112
SPLITTER_#1(after.combustor)
MRc12

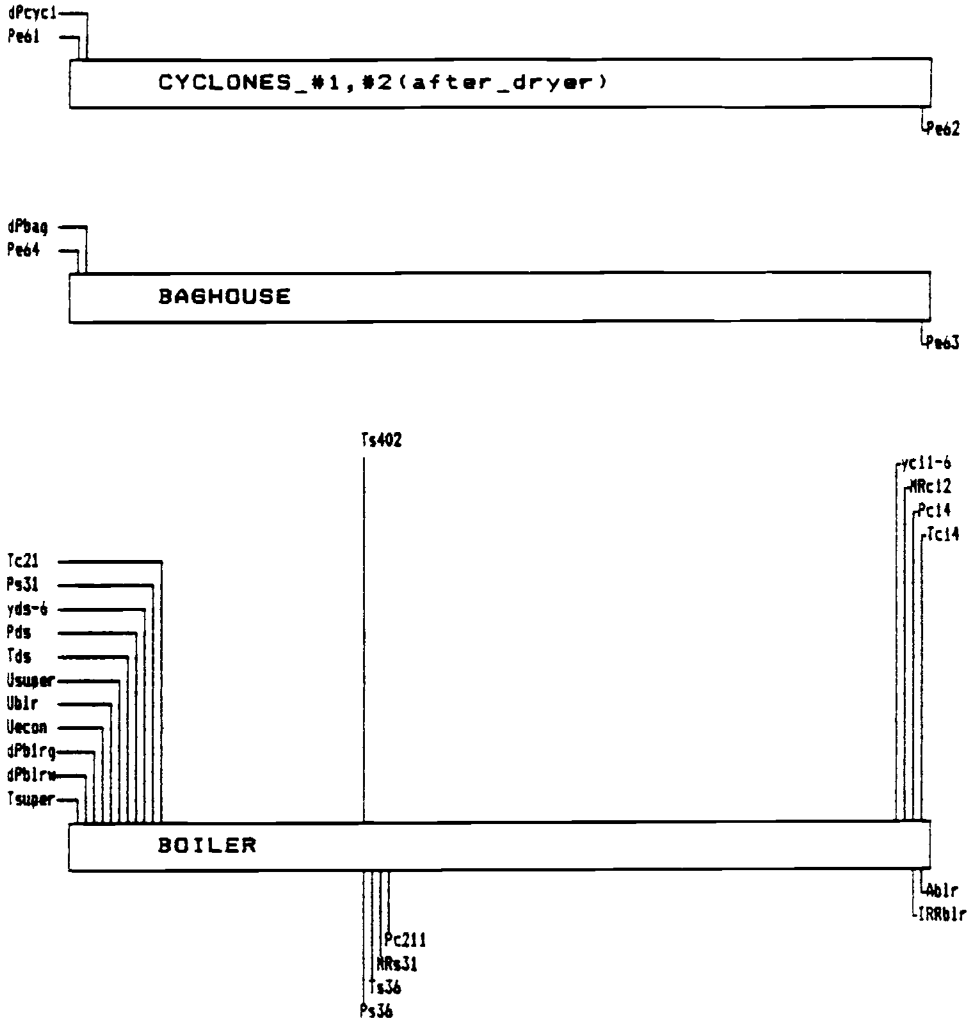

Ts36
Ps36
MRs31
yc11-6
Pc113
MRc12
Tc112
Ts38
SnDhxi
SpDhxi
chxi
xhxi
nwhxi
npahxi
nppaxi
typhxi
fshxi
rufhxi
yds-6
Pds
Tds
fouhxi
Kphxi
ODhxi
IDhxi
STEAM-GAS_HEAT_EXCHANGER
dPhxiw
dPhxig
Ahxi
Vhxig
Pc13
Tc13
Ps38
IRRhxi

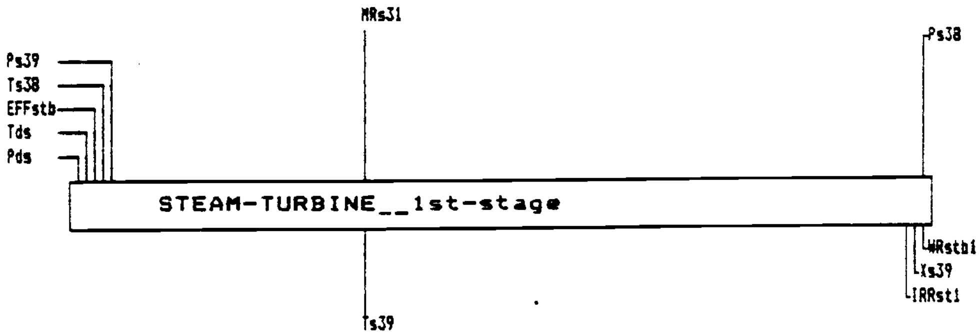

Pa022
Ta022
ya02-6
ycl1-6
MRc12
Pc13
Tc13
MRa022
Ta031
SaDhx2
SpDhx2
chx2
xhx2
nuhx2
npahx2
npphx2
typhx2
fshx2
rufhx2
yds-6
Pds
Tds
fouhx2
Kphx2
ODhx2
IDhx2
AIR-GAS_HEAT_EXCHANGER
dPhx2a
dPhx2g
Ahx2
Vhx2g
Tc14
Pc14
IRRhx2
Pa031

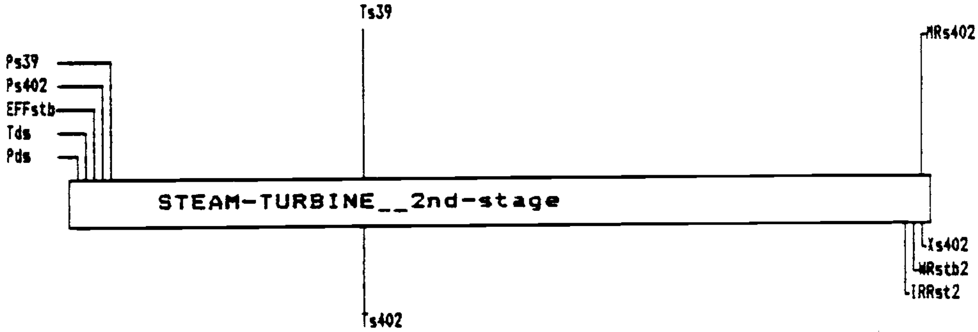

Tc26
MRc25
ycl1-6
Pc113
MRc16a
Tc112
Tc21
Pc22
yds-6
Pds
Tds
MIXER(to.dryer)
Pc26
MRc26
IRRmxd
MRc16

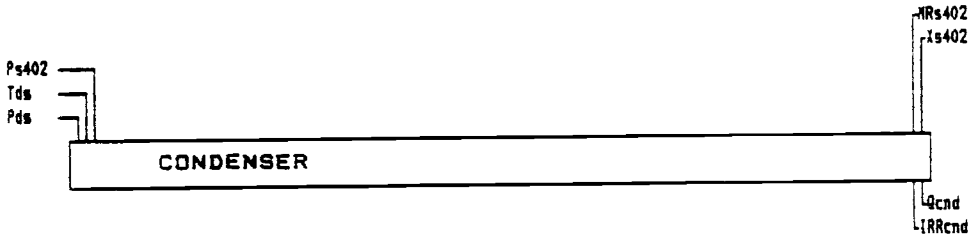

Pc29
dPapl
AIR_POLLUTION_DEVICE.#1
Pc27

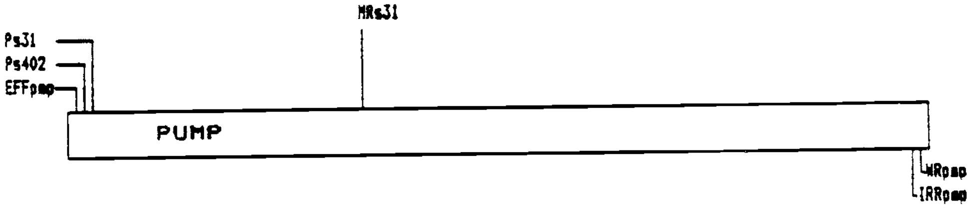

yc11-6
MRc26
MRb52
MRbdwd
Ta01
Wb51
Wb52
Te61
Pe61
yds-6
Pds
Tds
dPdry
drylos
HHV
DRYER
MRb51
MRe61
ye61-6
lendry
WRdry
IRRdry
Tc26
Pc29

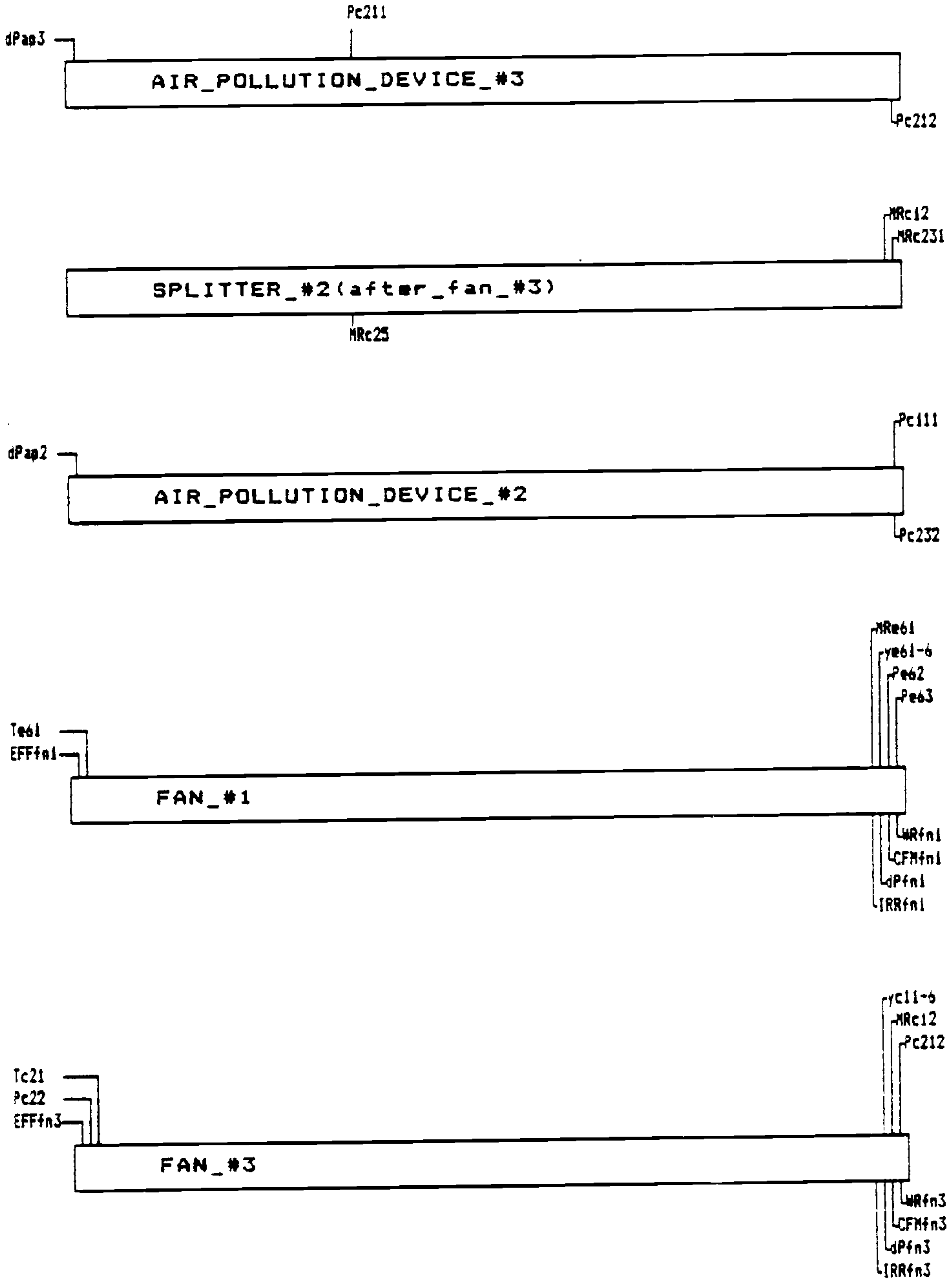

dPcycl
Pe61
CYCLONES_#1,#2(after_dryer)
Pe62
dPbag
Pe64
BAGHOUSE
Pe63
Ts402
ycl1-6
MRc12
Pc14
Tc14
Tc21
Ps31
yds-6
Pds
Tds
Usuper
Ublr
Uecon
dPblrg
dPblrw
Tsuper
BOILER
Ablr
IRRblr
Pc211
MRs31
Ts36
Ps36

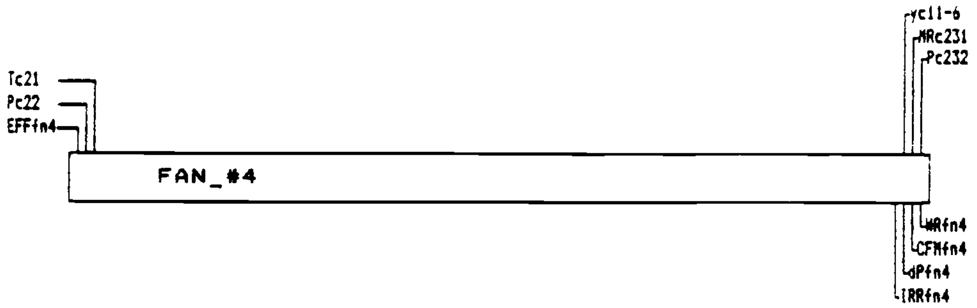

MRs31
Ps38
Ps39
Ts38
EFFstb
Tds
Pds
STEAM-TURBINE__1st-stage
WRstb1
Xs39
IRRst1
Ts39

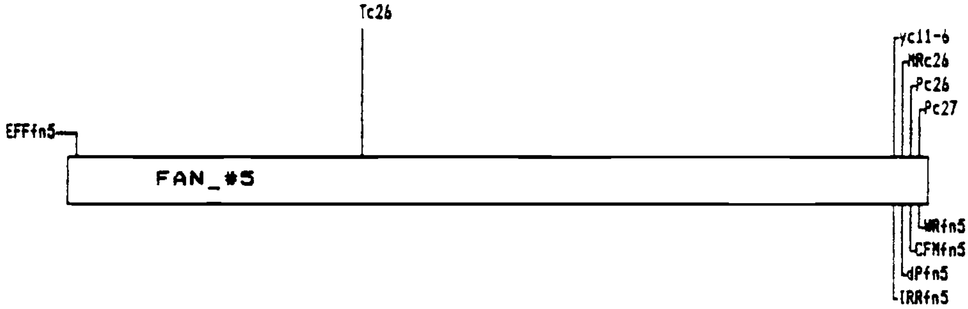

Ts39
MRs402
Ps39
Ps402
EFFstb
Tds
Pds
STEAM-TURBINE__2nd-stage
Xs402
WRstb2
IRRst2
Ts402

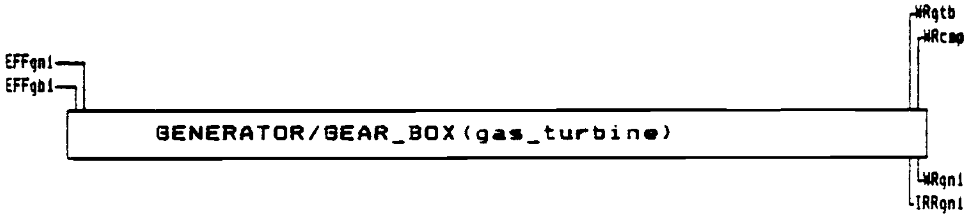

MRs402
Xs402
Ps402
Tds
Pds
CONDENSER
Qcnd
IRRcnd

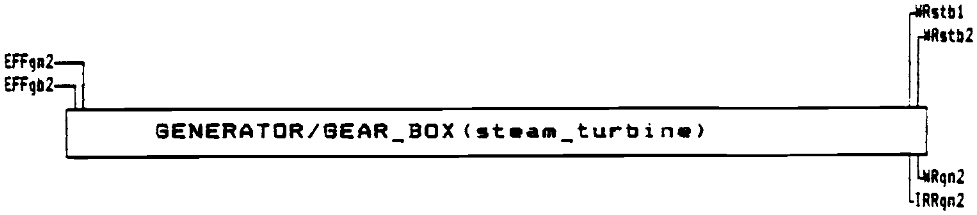

MRs31
Ps31
Ps402
EFFpmp
PUMP
WRpmp
IRRpmp

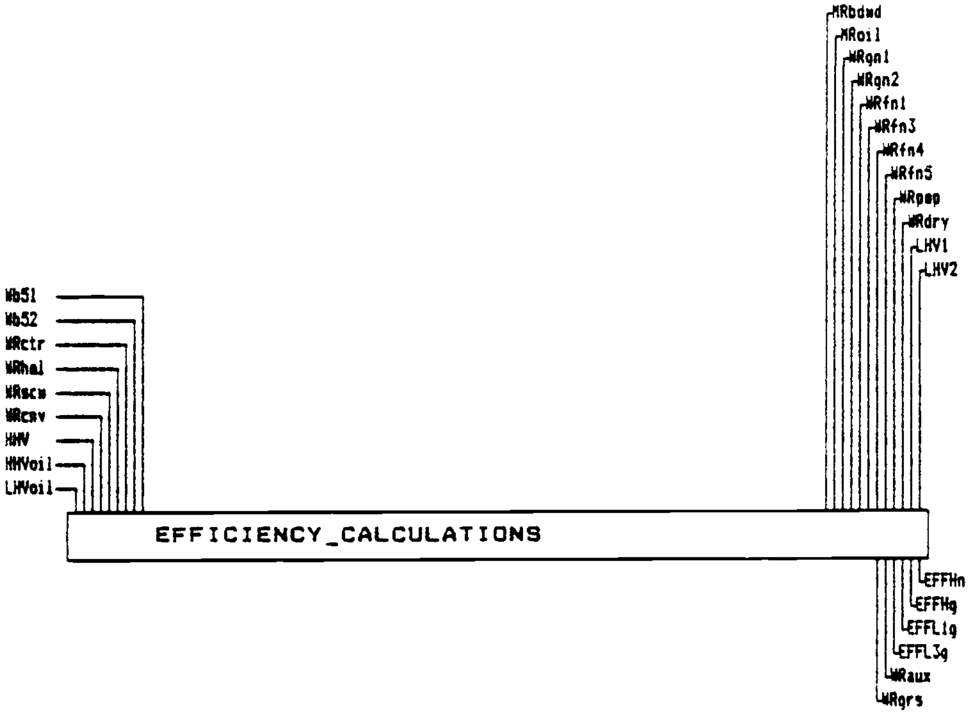

Pc211
dPap3
AIR_POLLUTION_DEVICE_#3
Pc212
MRc12
MRc231
SPLITTER_#2(after_fan_#3)
MRc25
Pc111
dPap2
AIR_POLLUTION_DEVICE_#2
Pc232
MRe61
ye61-6
Pe62
Pe63
Te61
EFFfn1
FAN_#1
WRfn1
CFMfn1
dPfn1
IRRfn1
yc11-6
MRc12
Pc212
Tc21
Pc22
EFFfn3
FAN_#3
WRfn3
CFMfn3
dPfn3
IRRfn3


Tc21
Pc22
EFFfn4
FAN_#4
ycl1-6
MRc231
Pc232
WRfn4
CFMfn4
dPfn4
IRRfn4


Tc26
EFFfn5
FAN_#5
ycl1-6
MRc26
Pc26
Pc27
WRfn5
CFMfn5
dPfn5
IRRfn5


EFFgn1
EFFgb1
GENERATOR/GEAR_BOX(gas_turbine)
WRgtb
WRcmp
WRgn1
IRRgn1


EFFgn2
EFFgb2
GENERATOR/GEAR_BOX(steam_turbine)
WRstb1
WRstb2
WRgn2
IRRgn2



MRbdwd
MRoil
WRgn1
WRgn2
WRfn1
WRfn3
WRfn4
WRfn5
WRpmp
WRdry
LHV1
LHV2
Wb51
Wb52
WRctr
WRhal
WRscw
WRcmv
HHV
HHVoil
LHVoil
EFFICIENCY_CALCULATIONS
EFFHn
EFFHg
EFFL1g
EFFL3g
WRaux
WRgrs

APPENDIX B

An Example of Using PROSIM

Listing of PROSIM and EQNS

Sample Run from PROSIM for the Biomass Plant

Appendix B

An Example of Using PROSIM

A problem taken from Stoecker [1980, problem 6-10] is used in this example to show the entire sequence of modeling a thermal process. This example is not as complex as the Biomass project, and is used to present a reasonably short complete explanation. Each step mentioned in Chapter 3 is discussed below. The problem, as Stoecker presents it, is as follows:

> A helium liquefier operating according to the flow diagram shown in figure B.1 receives high-pressure helium vapor, liquefies a fraction of the vapor, and returns the remainder to be recycled. The following operating conditions prevail:
>
> Point 1 (vapor entering warm side of heat exchanger), T = 15 K, h = 78.3 kJ/kg, w = 5 g/s, p = 2000 kPa.
>
> Point 5 (vapor leaving turbine), T = 8 K, h = 53 kJ/kg, w = 4 g/s.
>
> Separator, p = 100 kPa, saturation temperature at 100 kPa = 4.2 K, hf = 10 kJ/kg, hg = 31 kJ/kg.
>
> Heat exchanger, UA = 100 W/K.
>
> Specific heat of helium vapor:
> Cp = 6.4 kJ/(kg-K) at 2000 kPa
> Cp = 5.8 kJ/(kg-K) at 100 kPa
>
> Simulate the system, determining the values of w4, T2, T7, and T8.

Step 1 — Decide what values are constant, and what values are unknown.

In this case, the unknowns and constants have been chosen by Stoecker in his problem statement. In most thermal processes, the choice is made through considerations of thermodynamics, experience,

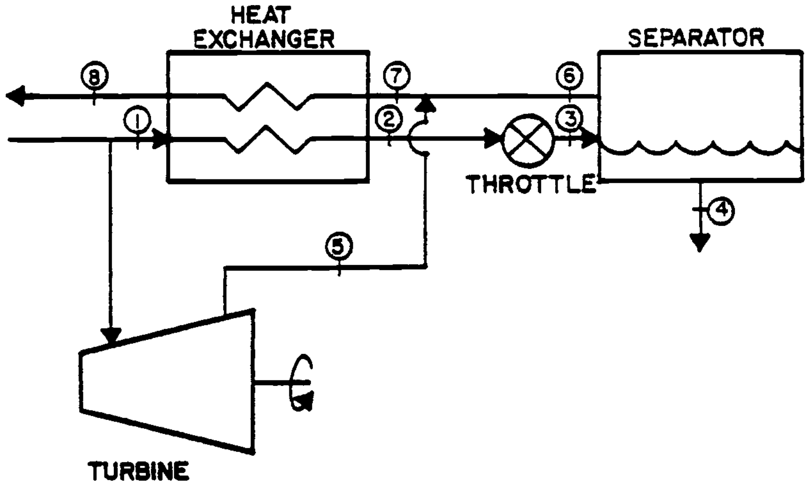


Figure B.1 Helium liquefier process, from Stoecker (1980).

and knowledge of the process being considered.

From the specified conditions at state points 1 and 5, the specification of the heat exchanger, and the known throttled pressure, the constants therefore are T1, MR1, P1, T5, MR5, UA, and P3.

In the heat exchanger, the exit temperatures and one of the flow rates are unknown. In the separator, both exit flow rates are unknown. In the mixer, the exit temperature is unknown. And in the throttle, the enthalpy is required in order to calculate the first law for the separator. Therefore the unknowns are T2, T8, MR7, MR4, MR6, T7, and h2.

Note that in his problem statement, Stoecker gives some thermodynamic property values. But in general, these are also unknown. They are calculated as the process is simulated. These include T6, hf and hg in the separator, and the enthalpies at state points 1 and 5. T6 will be considered an unknown in this simulation.

The choice of state point thermodynamic properties that will be considered as values to be kept, and those that will be calculated, is not important. However, they must form a set of independent properties such that the other dependent properties can be obtained with the proper equations or tables.

Step 2 — Determine the inputs and outputs for each component, and draw the information flow diagrams.

In the heat exchanger, if the inlet temperatures, both flow rates, and the UA value are known, the exit temperatures can be calculated. Therefore the inputs are T1, T7, MR1, MR7, P1, P3, and UA; the outputs are T2 and T8.

In the throttle, enthalpy is constant. No calculations are required. In this example, the pressure drop is calculated and printed, but is not an output of the model to any other component model.

The separator exit flow rates can be determined from energy and mass balances. For the two balances, the inlet flow rate, pressure, and enthalpy must be input. From this, due to the saturated condition in the separator, the saturation temperature, and the liquid and gaseous saturation enthalpies can be obtained from thermodynamic property data. Therefore the inputs are MR1, h2, and P3; the outputs are MR4, MR6, and T6.

The mixer also uses energy and mass balances. Its inputs are the flow rates and state point properties of the inlet streams, and the exit conditions are calculated. The inputs are MR5, T5, P3, MR6, and T6; the outputs are MR7 and T7.

The turbine inlet and exit conditions are completely specified. Although Stoecker does not ask for the power output, it is easily calculated, and is the only output. The inputs required are MR5, P1, T1, P3, and T5.

Now, each component's inputs and outputs have been determined. Some of the inputs are constants, others are variables, and the rest are intermediate calculations. The outputs may be variables, intermediate calculations, or printed results.

There is actually no need to have any intermediate calculations since they could be considered variables. However, guesses need to be made, and checks for convergence will be performed for each variable. Thus it is often easier to make some of the unknowns intermediate calculations. Those that simply exit one component, then enter the next component, are good candidates for intermediate calculations. h2 is such a candidate, since it is determined in the heat exchanger and then passed directly to the separator.

Some of the unknowns are not required in any other components; these can be printed in the routine that determines them. DeltaP, Power, MR4, T2, T6 and T8 are the printed results in this example. The other unknowns are variables. They are MR6, MR7, and T7. Any inputs not mentioned as being variables or intermediate calculations are constants.

The information flow diagrams are now ready to be drawn, since the models for each component have been determined (although how those models determine the outputs has not been established yet). Each component has its own module box, with the inputs and outputs drawn as lines into and out of the box. Constants enter from the top left, variables from the top center, and intermediate calculations from the top right. Printed results exit the right side, variables from the bottom middle, and intermediate calculations from the bottom right.

The flow diagrams are in figure B.2. Following is the input to FLOWDIAG.

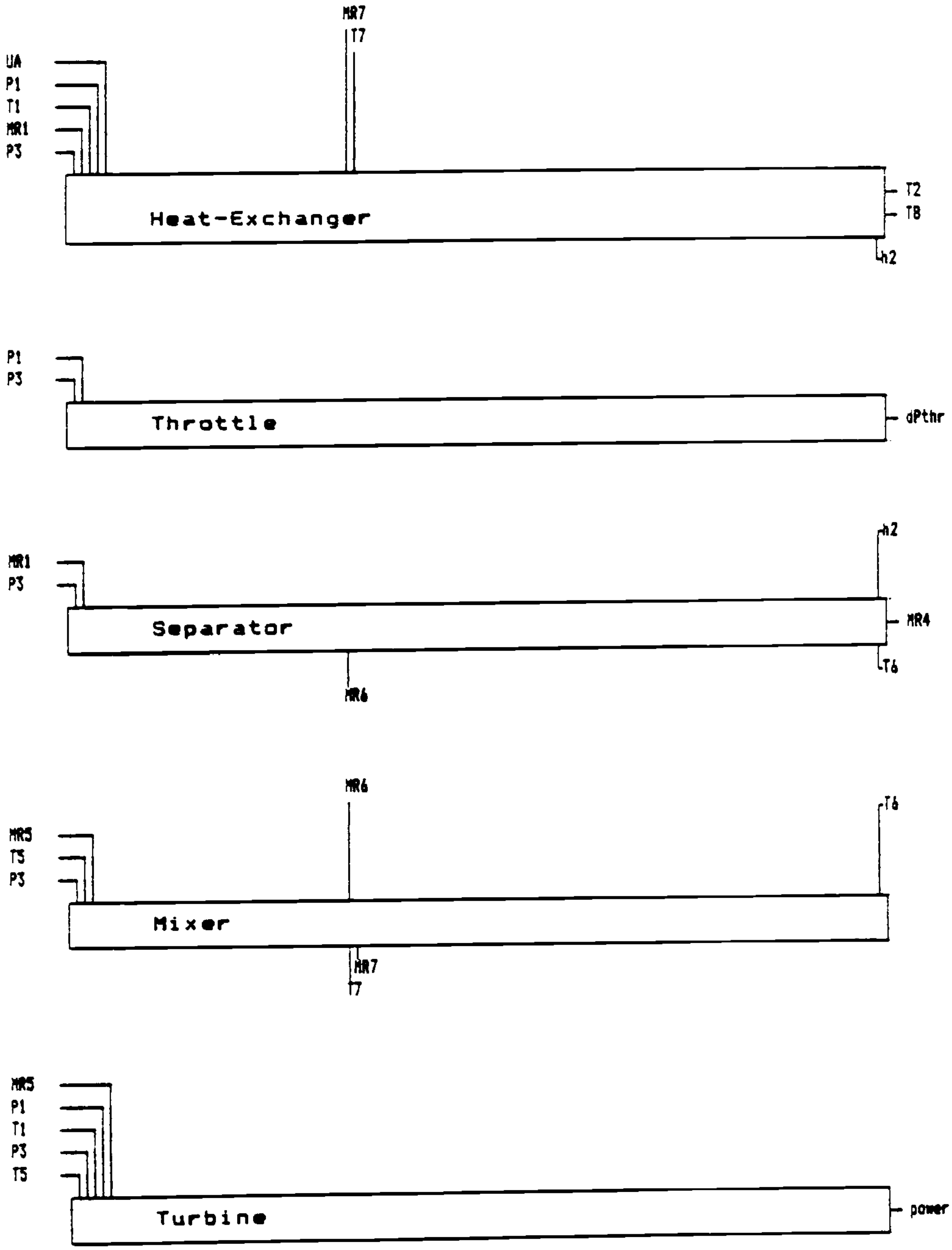


Figure B.2 Flow diagrams for Stoecker's problem.

```
box
   Heat-Exchanger
consin UA P1 T1 MR1 P3
varin MR7 T7
print T2 T8
calcex h2
box
   Throttle
consin P1 P3
print dPthr
box
   Separator
consin MR1 P3
calcin h2
varex MR6
calcex T6
print MR4
box
   Mixer
consin MR5 T5 P3
varin MR6
calcin T6
varex MR7 T7
box
   Turbine
consin MR5 P1 T1
       P3 T5
print power
```

Step 3 — Write the component subroutines.

The mathematical models for each component follow.

Heat Exchanger (see Stoecker [1980], pp 77-81)

$$w_a = MR_a \, C_p$$

$$w_b = MR_b \, C_p$$

$$D = UA \, (1/w_a - 1/w_b)$$

$$T_{a,out} = T_{a,in} - \frac{(T_{a,in} - T_{b,in}) \, (1 - e^D)}{(w_a/w_b - e^D)}$$

$$T_{b,out} = T_{b,in} + w_b/w_a \, (T_{a,in} - T_{a,out})$$

$h_{a,out} = h_{a,in} - C_p \ (T_{a,in} - T_{a,out})$

Separator

$MR_{vap} = MR_{in} \ (h_{in} - h_{liq}) \ / \ (h_{vap} - h_{liq})$

$MR_{liq} = MR_{in} - MR_{vap}$

Mixer

$MR_{out} = dMR_{in}$

$T_{out} = d(MR_{in} \ T_{in}) \ / \ MR_{out}$

Turbine

$Power = MR_{in} \ h_{in} - MR_{out} \ h_{out}$

Property Functions Required

h(T,P)
Cp(P)
hf(P)
hg(P)
Tsat(P)

The FORTRAN code for the component subroutines follow.

```
C========= C O M P O N E N T     M O D E L S ==================
C=========    Problem 6-10, Stoecker's Design of Thermal Systems
C                                        ------ -- ------- -------

      subroutine HxChr(Tain,Tbin, MRa,MRb, Pa,Pb, UA, haout,print)
      implicit real (a-z)
      logical*2 print
      Wa = MRa * CpHe(Pa)
      Wb = MRb * CpHe(Pb)
      D = UA * (1.0/Wa - 1.0/Wb)
      Taout = Tain - (Tain - Tbin) * (1.0 - exp(D)) / (Wa/Wb - exp(D))
      Tbout = Tbin + Wa/Wb * (Tain - Taout)
      haout = hHe(Tain, Pa) - CpHe(Pa) * (Tain - Taout)
      if (print) then
        write(8,10) '------ Heat Exchanger ------'
        write(8,10) 'Stream A: exit temperature = ',Taout,' K')
        write(8,10) 'Stream B: exit temperarure = ',Tbout,' K')
```

```
      endif
      return
   10 format(10x,a,f10.3,a)
      end

      subroutine Throtl(Pin, Pout, print)
      logical*2 print
      if (print) then
         write(8,10) '------ Throttle ------'
         write(8,10) 'Delta P = ', Pin-Pout, ' kPa'
      endif
      return
   10 format(10x,a,f10.3,a)
      end

      subroutine Separ(MRin, hin, P, MRvap, Tvap, print)
      implicit real (a-z)
      logical*2 print
      MRvap = MRin * (hin - hfHe(P)) / (hgHe(P) - hfHe(P))
      MRliq = MRin - MRvap
      Tvap = TsatHe(P)
      if(print) then
         write(8,10) '------ Separator ------'
         write(8,10) 'Liquid extracted = ',MRliq,' g/s'
         write(8,10) 'Temperature = ',TsatHe(P), 'K'
      endif
      return
   10 format(10x,a,f10.3,a)
      end

      subroutine Mixer(MRa, MRb, MRc, P, Ta, Tb, Tc, print)
C     ----- 'a' and 'b' are inlet streams; 'c' is outlet stream -----
      implicit real (a-z)
      logical*2 print
      MRc = MRa + MRb
      Tc = (MRa * Ta + MRb * Tb) / MRc
      if(print) write(8,10) '------ Mixer ------'
      return
   10 format(10x,a,f10.3,a)
      end

      subroutine Turbin(Tin, Pin, Tout, Pout, MR, print)
      implicit real (a-z)
      logical*2 print
      Power = MR * ( hHe(Tin,Pin) - hHe(Tout,Pout) )
      if(print) then
         write(8,10) '------ Turbine ------'
         write(8,10) 'Power output = ', Power, ' W'
      endif
      return
```

```
   10 format(10x,a,f10.3,a)
      end

C==========  P R O P E R T Y   F U N C T I O N S ===================
C==========                H E L I U M           ===================
C     ----- These are not intended to be complete, rather just -----
C     ----- examples of functions! Normally, curve fit equations -----
C     ----- would be in each function, along with domain checks. -----

      function hHe(T,P)
      if((T .eq. 15.0) .and. (P .eq. 2000.0)) hHe = 78.3
      if((T .eq.  8.0) .and. (P .eq.  100.0)) hHe = 53.0
      return
      end

      function CpHe(P)
      if (P .eq. 2000.0) CpHe = 6.4
      if (P .eq.  100.0) CpHe = 5.8
      return
      end

      function hfHe(P)
      if (P .eq. 100.0) hfHe = 10.0
      return
      end

      function hgHe(P)
      if (P .eq. 100.0) hgHe = 31.0
      return
      end

      function TsatHe(P)
      if (P .eq. 100.0) TsatHe = 4.2
      return
      end
```

Step 4 -- Write the EQNS subroutine.

The EQNS subroutine are coded with the help of the information flow diagrams. First, using the list of names, the array subscripts are assigned. One method is to sort the names alphabetically, then assign the number 1 to the first, 2 to the second, and so on.

There are three arrays: one for the constants, one for the variables, and one for the intermediate calculations. The assignments are made with data statements (which are more efficient since they are not executed each time EQNS is called).

Second, each module is called with its corresponding arguments. Those that are in the constants array are referred by C(subscript), the variables by V(subscript), and the intermediate calculations by R(subscript).

Finally, subroutine GETR lists the name and units of each intermediate calculation. It is used in the final printout. (Note that names and units for the variables and constants are entered through input data files.)

The FORTRAN code follows.

```
C============ EQNS ================================================
C  calculates the new variables as functionf of: the previous values of
C  the variables, and some constants.  Also returns intermediate calcs
C  to the calling routine to be printed along with the vars and consts.
C=================================================================
C  parameters:  c -- constant array; v -- variable array;
C               r -- result array;   title -- char string;
C               print -- print flag, if .TRUE., execute write stmts.
C=================================================================
      subroutine EQNS(c,v,r,title,print)
      implicit integer*2 (a-z)
      logical*2 print
      dimension c(*), v(*), r(*)
      real c,v,r
      character title*35
C     ------ the constants' subscripts ------
      data MR1, MR5, P1, P3, T1, T5, UA /1, 2, 3, 4, 5, 6, 7/
C     ------ the variables' subscripts -----
      data MR6, MR7, T7 /1, 2, 3/
C     ------ the results' subscripts ------
      data h2, T6 /1, 2/
C     ----- call the component subroutines -----
      if(print) write(8,10) title
      call HxChr (c(T1), v(T7), c(MR1), v(MR7), c(P1), c(P3), c(UA),
```

```
     >          r(h2), print)
      call Throtl(c(P1), c(P3), print)
      call Separ (c(MR1), r(h2), c(P3), v(MR6), r(T6), print)
      call Mixer (c(MR5), v(MR6), v(MR7), c(P3), c(T5), r(T6), v(T7),
     >          print)
      call Turbin(c(T1), c(P1), c(T5), c(P3), c(MR5), print)
      return
   10 format(10x,a)
      end


      subroutine GETR(rname, runits, numr)
      dimension rname(*), runits(*)
      character rname*6, runits*15
      rname(1) = 'h2    '
      runits(1) =       'J/g            '
      rname(2) = 'T6    '
      runits(2) =       'K              '
      numr = 2
      return
      end
```

Step 5 — Link PROSIM, EQNS and the component subroutines.

The various FORTRAN source files are compiled and linked together, using the appropriate software.

Step 6 — Prepare the input data files.

The constants are input from a data file. They are listed in the file in the same order as they were assigned in EQNS. (The order used here is alphabetic, convenient since the compiler lists the FORTRAN identifiers alphabetically, and FLOWDIAG lists the names that way. A sorting program can be used to order the input file.) The name and units for each constant are also contained in this file.

The guesses for the variables are input from another file. Again, they are listed in this file in the same order as they were assigned in EQNS (alphabetically). In addition, a minimum value, maximum value, and the name and units for each are listed.

The two data files are:

CONSTANTS DATA FILE

MR1	5.	g/s
MR5	4.	g/s
P1	2000.	kPa
P3	100.	kPa
T1	15.	K
T5	8.	K
UA	100.	W/K

VARIABLES DATA FILE

MR6	4.	0.	10.	g/s	.
MR7	9.	0.	20.	g/s	.
T7	5.97	0.	50.	K	.

Step 7 — Run the program.

The program requires several inputs from the user in addition to the two data files. For this problem, they were:

Debug flag -- F
Title -- Problem 6-10, Stoecker. 83/11/17
Tolerance -- 0.001
Maximum iterations -- 20

The results are listed on the next page.

Problem 6-10, Stoecker. 83/11/17

```
  8 iterations, specified tolerance:  .0010, actual tolerance:  .0003
----- Constants -----
MR1          5.0000 g/s              T1          15.0000 K
MR5          4.0000 g/s              T5           8.0000 K
P1        2000.0000 kPa              UA         100.0000 W/K
P3         100.0000 kPa
----- Variables -----
MR6          4.5554 g/s              T7           5.9767 K
MR7          8.5554 g/s
----- Results -----
h2          29.1327 J/g              T6           4.2000 K
```

Problem 6-10, Stoecker. 83/11/17

```
------ Heat Exchanger ------
Stream A: exit temperature =      7.318 K
Stream B: exit temperature =     10.930 K
------ Throttle ------
Delta P =   1900.000 kPa
------ Separator ------
Liquid extracted =        .445 g/s
Temperature =      4.200K
------ Mixer ------
------ Turbine ------
Power output =    101.200 W
```

```
$title:    'PROSIM -- process simulation program'
$storage:2
      program PROSIM
***************************************************************************
* PROSIM -- process simulation for non-linear, algebraic models.
* written by S. Fox  83.10.31
***************************************************************************
* IDENTIFIERS:
* c      -- the array of constants
* cname  -- the name of each constant
* cunits -- engineering units of each constant
* deltav -- the change in v from one iteration to another
* debug  -- if .TRUE., certain extra write statements are executed
* italld -- maximum allowable iterations
* iter   -- iteration number
* maxits -- if .TRUE., maximum iterations were attempted
* numc   -- number of constants
* numr   -- number of intermediate calculations
* numv   -- number of variables
* r      -- the array of intermediate calculations
* rold   -- previous values of the r array
* rname  -- the name of each intermediate calculation
* runits -- engineering units of each intermediate calculation
* solutn -- if .TRUE., solution has been found
* tolrnc -- maximum tolerance of the deltav's permitted
* v      -- the array of variables
* vmax   -- maximum allowable value of each variable
* vmin   -- minimum allowable value of each variable
* vname  -- the name of each variable
* vold   -- previous values of the v array
* vratio -- ratio of deltav to v
* vunits -- engineering units of each variable
* wayoff -- if .TRUE., one or more variables went out-of-bounds
* worst  -- worst of the vratios
***************************************************************************
      DIMENSION c(140), Cold(140), cunits(140), cname(140)
      DIMENSION v(50),vold(50),vmin(50),vmax(50),vunits(50),vname(50)
      DIMENSION r(160), rold(160), runits(160), rname(160)
      CHARACTER cname*6,rname*6,vname*6,cunits*15,runits*15,vunits*15
      CHARACTER big*1,page*1,title*35
      LOGICAL solutn,wayoff,maxits,debug
      pause 'Order of files changed: input 1st, output 2nd.'
C     ----- prompt user for debug flag -----
      write(*,100) 'Debug on/off? (T = extra output) Enter T or F:'
      read(*,110) debug
C     ----- obtain constants, initial guesses, and int calc names -----
      write(*,*) 'INPUT files:'
      write(*,*) '   Unit 4--constants.'
      write(*,*) '   Unit 5--variables.'
      write(*,*) 'OUTPUT files (with typical output file name):'
      write(*,*) '   Unit 1--warning file (CON).'
      write(*,*) '   Unit 6--constants, variables, intermediate calcs ',
     >                   'listing (PRN).'
```

```
      write(*,*) '   Unit 8--component subroutines output (PRN).'
      call GETC(c,Cold,cname,cunits,numc,debug)
      call GETV(v,vmin,vmax,vname,vunits,numv,debug)
      call GETR(rname,runits,numr)
      if(numc+numv .eq. 0) stop 'No input data, check input files ...'
C     ----- open output files -----
      open(1,file=' ',status='new')
      open(6,file=' ',status='new')
      open(8,file=' ',status='new')
C     ----- prompt user and initialize -----
      write(*,100) 'Enter title for output (max 35 characters):  '
      read(*,200) title
      write(*,100) 'Enter tolerance (e.g. 0.01):'
      read(*,210) tolrnc
      if (tolrnc .eq. 0.0) tolrnc = 0.01
      write(*,100) 'Enter max iterations allowed (e.g. 20):'
      read(*,220) italld
      if (italld .eq. 0) italld = 20
      iter = -1
      maxits = .FALSE.
      wayoff = .FALSE.
C========== repeat ==================================================
   10 continue
C       ----- initialize for this iteration -----
        iter = iter + 1
        solutn = .TRUE.
        do 15 i = 1, numv
   15     vold(i) = v(i)
        do 16 i = 1, numr
   16     rold(i) = r(i)
C       ----- perform the equations to get new v's -----
        call EQNS(c,v,r,title,debug)
        write(*,1010) 'Iteration ',iter
C       ----- check for convergence -----
        call CHKCON(vold,v,vname,numv,tolrnc,solutn,worst)
        call CHKCON(rold,r,rname,numr,tolrnc,solutn,worst)
        if( .not. solutn) then
C         ----- check for max iterations -----
          if( iter .eq. italld) then
            maxits = .TRUE.
            DEBUG = .true.
          endif
C         ----- check for variables out-of-bounds -----
          do 40 i = 1, numv
            if( (v(i) .lt. vmin(i)) .or. (v(i) .gt. vmax(i)) ) then
              write(*,1015)  vname(i),' out of allowable bounds'
              wayoff = .TRUE.
              debug = .TRUE.
            endif
   40     continue
        endif
        if(wayoff) pause
C       ----- write out current values of each variable -----
```

```
        if(debug) then
          call deltac(Cold,c,cname,cunits,numc,0)
          call RESULT('----- Variables -----',v,vname,vunits,numv,0)
          call RESULT('----- Int.Calcs -----',r,rname,runits,numr,0)
        endif
C========== until solution or max-iterations or way-off ============
      if ( (.not. solutn) .and. (.not. maxits) .and. (.not. wayoff) )
     >  goto 10
C     ----- printer control chars: double wide & new page -----
      big = char(14)
      page = char(12)
C     ----- send results of simulation to output device -----
      if( .not. debug) then
        write(*,101) 'One more iteration to print in subroutines'
        iter = iter + 1
        do 50 i = 1, numv
   50     vold(i) = v(i)
        do 51 i = 1, numr
   51     rold(i) = r(i)
        call EQNS(c,v,r,title,.TRUE.)
C       ----- check convergence, assume won't go out of bounds -----
        call CHKCON(vold,v,vname,numv,tolrnc,solutn,worst)
      endif
C     ----- send constants, variables, and int calcs to output device ---
      write(6,'(''1''/10x,a,a)') big,title
      write(6,1110) iter, ' iterations,'
      write(6,1114) ' specified tolerance: ',tolrnc
      write(6,1115) ', actual tolerance: ', worst
      call RESULT('----- Constants -----',c,cname,cunits,numc,6)
      call deltac(Cold,c,cname,cunits,numc,6)
      write(6,'(1x,a,a,a)') page,big,title
      call RESULT('----- Variables -----',v,vname,vunits,numv,6)
      write(6,'(1x,a,a,a)') page,big,title
      call RESULT('----- Int.Calcs -----',r,rname,runits,numr,6)
      call deltac(Cold,c,cname,cunits,numc,0)
      write(6,'(1x,a)')
C     ----- send out a closing comment -----
      if( solutn )  write(*,*) 'VALID SOLUTION'
      if( wayoff )  write(*,*) 'SOLUTION DIVERGED'
      if( maxits )  write(*,*) 'DID NOT CONVERGE'
C     ----- you're all done, close up shop -----
      close(1)
      close(6)
      close(8)
C     ----- Want a Lotus 1-2-3 readable output file? -----
      call Lotus(c,cname,cunits,numc, v,vname,vunits,numv,
     >           r,rname,runits,numr)
      stop
C     ----- formats -----
  100 format(1x,a\)
  101 format(1x,a )
  110 format(L1)
  200 format(bn,a)
```

```
  210 format(bn,f10.0)
  220 format(bn,i10)
 1010 format(1x,a,i3)
 1015 format(1x,'=======> ',a,a)
 1020 format(1x,a,a,f10.4,1x,a\)
 1021 format(1x,a,a,f10.4,1x,a )
 1110 format(/10x,i3,a\)
 1114 format(a,f6.4\)
 1115 format(a,f6.4)
      end

      subroutine GETC (c,Cold,cname,cunits,numc,print)
*****************************************************************************
* GETC   -- obtains the constants, their name and their units
*           initializes the Cold array
*           print -- if .TRUE., write constants
*****************************************************************************
      DIMENSION c(*),Cold(*),cname(*),cunits(*)
      CHARACTER cname*6,cunits*15
      LOGICAL*2 print
      open(4,file=' ')
      if(print) write(*,200) ' Reading constants '
      do 10 i = 1, 32767
         read(4,100,end=20) cname(i),c(i),cunits(i)
         Cold(i) = c(i)
         if(print) write(*,200) '.'
   10 continue
   20 continue
      close(4)
      numc = i - 1
      if(print) then
         write(*,'(i3)') numc
         call RESULT('Constants:           ',c,cname,cunits,numc,0)
      endif
      return
  100 format(bn,a6,4x,f10.0,a15)
  200 format(a\)
      end

      subroutine GETV (v,vmin,vmax,vname,vunits,numv,print)
*****************************************************************************
* GETV   -- obtains the guesses for the variables.
*           Also gets the minimum and maximum values for the variables,
*           the name and units of the variables,
*           and the number of variables.
*           print -- if .TRUE., then write guesses.
*****************************************************************************
      DIMENSION v(*),vmin(*),vmax(*),vname(*),vunits(*)
      CHARACTER vname*6,vunits*15
      LOGICAL*2 print
      open(5,file=' ')
      if(print) write(*,200) ' Reading initial guesses '
      do 10 i = 1, 32767
```

```
         read(5,100,end=20) vname(i), v(i), vmin(i), vmax(i), vunits(i)
         if(print) write(*,200) '.'
   10 continue
   20 continue
      close(5)
      numv = i - 1
      if(print) then
         write(*,'(i3)') numv
         call RESULT('Initial guesses:     ',v,vname,vunits,numv,0)
      endif
      return
  100 format(a6,4x,3f10.0,bn,a15)
  200 format(a\)
      end

      subroutine CHKCON(vold,v,vname,numv,tolrnc,solutn,worst)
C-----------------------------------
C  CHKCON -- checks for convergence
C-----------------------------------
      dimension vold(*),v(*),vname(*)
      character*6 vname
      logical*2 solutn
      worst = 0.0
      do 30 i = 1, numv
        deltav = abs(vold(i) - v(i))
C       ----- for large numbers, i.e. |v| > tolerance -----
        if( abs(v(i)) .gt. tolrnc) then
          vratio = abs( deltav / v(i) )
          if(vratio .gt. tolrnc) then
            solutn = .FALSE.
          endif
          if (vratio .gt. worst) then
            worst = vratio
            write(*,1) vname(i),worst
          endif
C       ----- for small numbers, i.e. |v| <= tolerance -----
        else
          if(deltav .gt. tolrnc) then
            solutn = .FALSE.
          endif
          if (deltav .gt. worst) then
            worst = deltav
            write(*,1) vname(i),worst
          endif
        endif
   30 continue
      write(*,'(a)') ' '
      return
    1 format(2x,a,':',f5.3\)
      end

      subroutine RESULT(string,x,xname,xunits,numx,lun)
*****************************************************************************
```

```
* RESULT -- prints the identifier x, its name, and units.
*          The arrays are printed in two columns, from top to bottom.
* written by S. Fox 83.07.15
************************************************************************
      DIMENSION x(*),xname(*),xunits(*)
      CHARACTER xname*6,xunits*15,string*21
      LOGICAL*2 even
      write(lun,90) string
      lines = numx / 2
C     ----- check for even or odd number of x's -----
      if( lines * 2 .eq. numx) then
        even = .TRUE.
      else
        even = .FALSE.
      endif
C     ----- write the two columns -----
      do 10 i = 1, lines
        write(lun,100) xname(i), x(i), xunits(i)
        if (even) then
          write(lun,101) xname(i+lines), x(i+lines), xunits(i+lines)
        else
          write(lun,101) xname(i+lines+1),x(i+lines+1),xunits(i+lines+1)
        endif
   10 continue
C     ----- if odd, write the middle x -----
      if( .not. even) then
        write(lun,102) xname(lines+1), x(lines+1), xunits(lines+1)
      endif
      return
C     ----- formats -----
   90 format(10x,a)
  100 format(10x,a,1x,f10.4,1x,a\)
  101 format( 1x,a,1x,f10.4,1x,a)
  102 format(10x,a,1x,f10.4,1x,a)
      end

      subroutine deltac(c,Cold,cname,cunits,numc,lun)
*-----------------------------------------
*  checks to see if any constants were changed
*-----------------------------------------
      character cname*6,cunits*15
      dimension c(*),Cold(*),cname(*),cunits(*)
      do 10 i = 1,numc
   10   if(c(i) .ne. Cold(i)) write(lun,100) cname(i),c(i),Cold(i)
  100 format(10x,a,' was changed from ',f15.5,' to ',f15.5)
      return
      end

      subroutine Lotus(c,cname,cunits,numc, v,vname,vunits,numv,
     >                 r,rname,runits,numr)
C-----------------------------------------------------------------
C       Generate a file that Lotus 1-2-3 can read.
C       Sort the c, v, and r arrays as one large array.
```

```
C-----------------------------------------------------------------
      dimension c(*),cname(*),cunits(*), v(*),vname(*),vunits(*),
     >          r(*),rname(*),runits(*)
      dimension string(400)
      character cname*6,cunits*15, vname*6,vunits*15, rname*6,runits*15
      character string*40,temp*40, filnam*12
      logical*2 WantIt, InOrdr
C     ----- prompt for decision whether to make Lotus file -----
      write(*,9) 'Want output that can be imported into Lotus 1-2-3?'
      write(*,1) 'If so, enter T.  If not, enter F.  :'
      read(*,2) WantIt
      if (.not. WantIt) return
      numtot = numc + numv + numr
      if (numtot .gt. 400) then
         write(*,1) 'Hey, your arrays are too big, >400 total'
         return
      endif
C     ----- write c, v, and r arrays to string array -----
      write(*,1) 'Copying ...'
      do 10 i = 1, numc
         write(string(i), 6) cname(i), c(i), cunits(i)
   10 continue
      do 20 i = 1, numv
         write(string(i+numc), 6) vname(i), v(i), vunits(i)
   20 continue
      do 30 i = 1, numr
         write(string(i+numc+numv), 6) rname(i), r(i), runits(i)
   30 continue
C     ----- sort string array using Shell's algorithm -----
      write(*,1) 'Sorting ...'
      ndelta = numtot
   50 if (ndelta .gt. 1) then
         ndelta = ndelta / 2
   60    InOrdr = .true.
         do 70 i = 1, numtot-ndelta
            if (string(i) .gt. string(i+ndelta)) then
               temp = string(i)
               string(i) = string(i+ndelta)
               string(i+ndelta) = temp
               InOrdr = .false.
            endif
   70    continue
         if (.not. InOrdr) goto 60
         goto 50
      endif
C     ----- write to file -----
      write(*,1) 'Output file name (use extension .PRN):'
      read(*,3) filnam
      open(10,file=filnam,status='new')
      do 80 i = 1, numtot
         write(10,3) string(i)
   80 continue
      close(10)
```

```
      return
    1 format(1x,a\)
    2 format(L1)
    3 format(a)
    6 format(1h",a,1h",1x, e12.5,1x, 1h",a,1h")
    9 format(1x,a)
      end
```

```
$title: 'Biomass.  Equations subroutine'
$nolist
$storage:2
      subroutine EQNS(c,v,r,title.print)
      implicit integer*2 (a-z)
      logical*2 print
      real      c(*),v(*),r(*)
      character title*35,page*1,big*1
**********************************************************************
*  EQNS -- Calls the component subroutines.  EQNS is called by PROSIM.
*          Sends print flag to the component subroutines.
**********************************************************************
*  PARAMETERS:
*  c       -- constant array
*  v       -- variable array
*  r       -- result (intermediate calculation) array
*  title   -- a 35 character heading printed at the top of each page.
*  print   -- if .TRUE., then execute certain write statements.
**********************************************************************
$page
C     ----- initialize the subscripts for the c array -----
      DATA AFrat,
     >     chx1,chx2, cmpbld, dPap1,dPap2,dPap3,dPbag,dPblrg,
     >     dPblrw,dPcmb,dPcmp,dPcyc1,dPcyc3,dPdry,dPgtb,dPtrim,
     >     drylos, EFFcmp,EFFfn1,EFFfn3,
     >     EFFfn4,EFFfn5,EFFgb1,EFFgb2,EFFgn1,EFFgn2,EFFgtb,EFFpmp,
     >     EFFstb, fouhx1,fouhx2, fshx1,fshx2, HHV,HHVoil, IDhx1,IDhx2,
     >     Kphx1,Kphx2, LHVoil, MRc16m,MRc23m, MRstp,
     >     npnhx1,npnhx2,npphx1,npphx2, nuhx1,nuhx2, ODhx1,ODhx2
     >     / 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,
     >      23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,
     >      43,44,45,46,47,48,49,50,51,52/
      DATA Pa01,Pa04,Pc22, Pds, Pe61,Pe64, Prcinj, Prcmp,
     >     Ps31,Ps39,Ps402, radlos, radtrm, rufhx1,rufhx2,
     >     SnDhx1,SnDhx2,SpDhx1,SpDhx2,
     >     Ta01,Ta031,Ta032, Tc112,Tc21,
     >     Tds, Te61,Ts38, Tsuper, typhx1,typhx2, Ublr,Uecon, unbcbn,
     >     Usuper, Wa01,Wb51,Wb52, WRcnv,WRctr,WRhml,WRscw,
     >     Xash,Xashoi, XC, XCCO, XCCOoi, XCoil, Xdirt, XH2, XH2oil,
     >     xhx1,xhx2, XN2, XN2oil, XO2, XO2oil,
     >     ydsCD,ydsW,ydsO2,ydsN2,ydsAr,ydsCO
     >     / 53,54,55,56,57,58,59,60,61,
     >      62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,
     >      81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,
     >      100,101,102,103,104,105,106,107,108,109,110,111,112,113,114/

C     ----- initialize the subscripts for v array -----
      DATA MRc16, MRc25, MRs31, Pa031, Pc211, Pc29,Ps36,
     >     Tc26,Ts36,Ts39,Ts402
     >     /1,2,3,4,5,6,7,8,9,10,11/

C     ----- initialize the subscripts for r array (int. calcs) -----
      DATA Ablr,Ahx1,Ahx2,
```

```
     >     CFMfn1,CFMfn3,CFMfn4,CFMfn5,dPfn1,dPfn3,dPfn4,dPfn5,
     >     dPhx1g, dPhx1w, dPhx2a, dPhx2g,
     >     EFFHg,EFFHn,EFFL1g,EFFL3g,
     >     IRRblr,IRRcmb,IRRcmp,IRRcnd,IRRdry,IRRfn1,IRRfn3,IRRfn4,
     >     IRRfn5,IRRgn1,IRRgn2,IRRgtb,IRRhx1,IRRhx2,IRRinj,IRRmxc,
     >     IRRmxd,IRRpmp,IRRst1,IRRst2,IRRtrm,
     >     lendry,LHV1,LHV2,
     >     Ma01, Ma02, Ma03
     >     /1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,
     >      23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,
     >      38,39,40,41,42,43,44,45,46/

      DATA MRa021,MRa022,MRa032,MRa04,MRa06,MRb51,MRb52,MRbdwd,
     >     MRc111,MRc112,MRc12,MRc231,MRc26,MRe61, MRoil, MRs402,
     >     Pa021,Pa022,Pa032,Pc111,Pc112,Pc113,Pc13,Pc14,Pc212,Pc232,
     >     Pc26,Pc27,Pe62,Pe63,Ps38, Qcmb,Qcnd,
     >     Ta021,Ta022,Ta04,Tc111,Tc13,Tc14, Vhx1g,Vhx2g,
     >     WRaux,WRcmp,WRdry,
     >     WRfn1,WRfn3,WRfn4,WRfn5,WRgn1,WRgn2,WRgrs,WRgtb,WRpmp,
     >     WRstb1,WRstb2, Xs39, Xs402
     >     /47,48,49,50,51,52,53,54,55,56,
     >      57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,
     >      76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,
     >      95,96,97,98,99,100,101,102,103/
      DATA ya01CD,ya01W,ya0102,ya01N2,ya01Ar,ya01CO,
     >     ya02CD,ya02W,ya0202,ya02N2,ya02Ar,ya02CO,
     >     ya03CD,ya03W,ya0302,ya03N2,ya03Ar,ya03CO,
     >     ya04CD,ya04W,ya0402,ya04N2,ya04Ar,ya04CO,
     >     yc11CD,yc11W,yc1102,yc11N2,yc11Ar,yc11CO,
     >     ye61CD,ye61W,ye6102,ye61N2,ye61Ar,ye61CO
     >     /104,105,106,107,108,109,110,111,112,113,114,115,
     >      116,117,118,119,120,121,122,123,124,125,126,127,128,129,
     >      130,131,132,133,134,135,136,137,138,139/
$page
      page = char(12)
      big =  char(14)
C     ----- call the component subroutines -----

      if(print) write(8,14) big,title

      call COMPRE( c(Ta01), c(Pa01), c(dPcmp), r(MRa021), r(MRa06),
     >     c(MRstp), r(Ta021), r(Pa021), r(WRcmp), c(EFFcmp),
     >     c(Prcmp), c(cmpbld),
     >     r(ya01CD),r(ya01W),r(ya0102),r(ya01N2),r(ya01Ar),r(ya01CO),
     >     r(Ma01), c(Wa01), c(Tds), c(Pds),
     >     c(ydsCD), c(ydsW), c(yds02), c(ydsN2), c(ydsAr), c(ydsCO),
     >     r(IRRcmp), print )

      call INJECT( v(MRs31), v(Ts39), c(Ps39), r(MRs402), r(MRa021),
     >     r(ya01CD),r(ya01W),r(ya0102),r(ya01N2),r(ya01Ar),r(ya01CO),
     >     r(Ma01), r(Ta021), r(Pa021), r(MRa022),
     >     r(ya02CD),r(ya02W),r(ya0202),r(ya02N2),r(ya02Ar),r(ya02CO),
     >     r(Ma02), c(Prcinj), r(Ta022), r(Pa022), c(Tds), c(Pds),
```

```
>     c(ydsCD), c(ydsW), c(yds02), c(ydsN2), c(ydsAr), c(ydsCO),
>     r(IRRinj), print )

 if(print .and. c(Ta031) .ne. c(Ta032)) write(8,12) page,big,title

 call TRIM( c(HHVoil), c(radtrm), c(Xashoi), c(XCoil), c(XH2oil),
>     c(XN2oil), c(XO2oil), c(XCCOoi), c(Ta032),
>     r(ya02CD),r(ya02W),r(ya0202),r(ya02N2),r(ya02Ar),r(ya02CO),
>     c(ydsCD), c(ydsW), c(yds02), c(ydsN2), c(ydsAr), c(ydsCO),
>     c(Tds), c(Pds), v(Pa031), c(dPtrim), c(Ta031), r(MRa022),
>     c(Ta01),
>     r(ya03CD),r(ya03W),r(ya0302),r(ya03N2),r(ya03Ar),r(ya03CO),
>     r(MRoil), r(MRa032), r(Pa032), r(Ma03), r(IRRtrm), print)

 if(print) write(8,12) page,big,title

 call GASTUR( c(Ta032), r(Pa032), r(MRa032),
>     r(Ta021), r(Pa021), r(MRa06),
>     r(Ta04), c(Pa04), r(MRa04), c(dPgtb), r(WRgtb), c(EFFgtb),
>     r(ya03CD),r(ya03W),r(ya0302),r(ya03N2),r(ya03Ar),r(ya03CO),
>     r(Ma03),
>     r(ya01CD),r(ya01W),r(ya0102),r(ya01N2),r(ya01Ar),r(ya01CO),
>     r(Ma01),
>     r(ya04CD),r(ya04W),r(ya0402),r(ya04N2),r(ya04Ar),r(ya04CO),
>     c(Tds), c(Pds),
>     c(ydsCD),c(ydsW),c(yds02),c(ydsN2),c(ydsAr),c(ydsCO),
>     r(IRRgtb), print )

 call BURNER( c(HHV), c(radlos), c(Xash), c(Xdirt), c(XC), c(XH2),
>     c(XN2), c(X02), c(Wb52), c(unbcbn), c(XCCO), c(AFrat),
>     r(ya04CD),r(ya04W),r(ya0402),r(ya04N2),r(ya04Ar),r(ya04CO),
>     c(ydsCD), c(ydsW), c(yds02), c(ydsN2), c(ydsAr), c(ydsCO),
>     c(Tds), c(Pds), c(Pa04), c(dPcmb),
>     r(Ta04), r(MRa04), c(Ta01),
>     r(yc11CD),r(yc11W),r(yc1102),r(yc11N2),r(yc11Ar),r(yc11CO),
>     r(LHV1), r(LHV2), r(MRb52), r(MRbdwd), r(MRc111), r(Tc111),
>     r(Pc111), r(Qcmb), r(IRRcmb), print )

 call MIXER ( 'after combustor    ', r(Tc111), r(Pc111),
>     r(MRc111), c(Tc21), r(Pc111), r(MRc231), c(Tc112), r(Pc112),
>     r(MRc112), c(MRc23m),
>     r(yc11CD),r(yc11W),r(yc1102),r(yc11N2),r(yc11Ar),r(yc11CO),
>     c(Tds), c(Pds),
>     c(ydsCD), c(ydsW), c(yds02), c(ydsN2), c(ydsAr), c(ydsCO),
>     r(IRRmxc), print )

 call DELTAP( 'Cyclone #3         ', r(Pc112), c(dPcyc3),
>              r(Pc113), print )

 call SPLITT( 'after combustor    ', r(MRc112), v(MRc16),
>     r(MRc12), print )

 if(print) write(8,12) page,big,title
```

```
call STMHX ( v(Ts36), v(Ps36), v(MRs31), c(Ts38), r(Ps38),
>    c(Tc112), r(Pc113), r(MRc12),
>    r(yc11CD),r(yc11W),r(yc1102),r(yc11N2),r(yc11Ar),r(yc11CO),
>    r(Tc13), r(Pc13), r(dPhx1w), r(dPhx1g), c(Tds), c(Pds),
>    c(ydsCD), c(ydsW), c(yds02), c(ydsN2), c(ydsAr), c(ydsCO),
>    c(ODhx1),c(IDhx1),c(Kphx1),c(SnDhx1),c(SpDhx1), c(chx1),
>    c(xhx1),
>    c(nuhx1), c(npnhx1), c(npphx1),c(typhx1),c(fshx1),c(fouhx1),
>    c(rufhx1), r(Ahx1), r(Vhx1g), r(IRRhx1), print )

if(print) write(8,12) page,big,title

call AIRHX ( r(Ta022), r(Pa022), r(MRa022),
>    r(ya02CD),r(ya02W),r(ya0202),r(ya02N2),r(ya02Ar),r(ya02CO),
>    c(Ta031), v(Pa031), r(Tc13), r(Pc13), r(MRc12),
>    r(yc11CD),r(yc11W),r(yc1102),r(yc11N2),r(yc11Ar),r(yc11CO),
>    r(Tc14), r(Pc14), r(dPhx2a), r(dPhx2g), c(Tds), c(Pds),
>    c(ydsCD), c(ydsW), c(yds02), c(ydsN2), c(ydsAr), c(ydsCO),
>    c(ODhx2),c(IDhx2),c(Kphx2), c(SnDhx2), c(SpDhx2), c(chx2),
>    c(xhx2),
>    c(nuhx2), c(npnhx2),c(npphx2),c(typhx2),c(fshx2),c(fouhx2),
>    c(rufhx2), r(Ahx2), r(Vhx2g), r(IRRhx2), print )

if(print) write(8,12) page,big,title

call MIXER ( 'to dryer            ', c(Tc21), c(Pc22), v(MRc25),
>    c(Tc112), r(Pc113), v(MRc16), v(Tc26), r(Pc26), r(MRc26),
>    c(MRc16m),
>    r(yc11CD),r(yc11W),r(yc1102),r(yc11N2),r(yc11Ar),r(yc11CO),
>    c(Tds), c(Pds),
>    c(ydsCD), c(ydsW), c(yds02), c(ydsN2), c(ydsAr), c(ydsCO),
>    r(IRRmxd), print )

call DELTAP( 'Air pollution dev 1 ', v(Pc29), -c(dPap1), r(Pc27),
>    print )

call DRYER ( r(MRb52), r(MRbdwd), c(HHV), r(MRc26), c(Pe61),
>    c(Te61), c(Wb52), c(Wb51), c(Ta01), c(drylos), -c(dPdry),
>    r(yc11CD),r(yc11W),r(yc1102),r(yc11N2),r(yc11Ar),r(yc11CO),
>    r(ye61CD),r(ye61W),r(ye6102),r(ye61N2),r(ye61Ar),r(ye61CO),
>    c(Tds), c(Pds),
>    c(ydsCD), c(ydsW), c(yds02), c(ydsN2), c(ydsAr), c(ydsCO),
>    v(Pc29), v(Tc26), r(MRb51), r(MRe61), r(WRdry), r(lendry),
>    r(IRRdry), print )

call DELTAP( 'Cyclones #1, #2     ', c(Pe61), c(dPcyc1),
>                r(Pe62), print )

call DELTAP( 'Bag house           ', c(Pe64), -c(dPbag),
>                r(Pe63), print )

call BOILER( r(Tc14), r(Pc14), r(MRc12),
```

```
     >    r(yc11CD),r(yc11W),r(yc1102),r(yc11N2),r(yc11Ar),r(yc11CO),
     >    c(Tc21), v(Pc211), v(Ts402), c(Ps31), v(MRs31), v(Ts36),
     >    v(Ps36), c(Tsuper), c(dPblrg), c(dPblrw), c(Usuper), c(Ublr),
     >    c(Uecon), c(Tds), c(Pds),
     >    c(ydsCD), c(ydsW), c(yds02), c(ydsN2), c(ydsAr), c(ydsCO),
     >    r(Ablr), r(IRRblr), print )

      if(print) write(8,12) page,big,title

      call STTURB( c(Ts38), r(Ps38), v(MRs31), c(EFFstb), v(Ts39),
     >    c(Ps39), r(WRstb1), r(Xs39), c(Tds), c(Pds), r(IRRst1),
     >    print )

      call STTURB( v(Ts39), c(Ps39), r(MRs402), c(EFFstb), v(Ts402),
     >    c(Ps402), r(WRstb2), r(Xs402), c(Tds), c(Pds), r(IRRst2),
     >    print )
c     ----- when no 2nd turbine is used, steam quality constant -----
      if( c(Ps39) .eq. c(Ps402) ) r(Xs402) = r(Xs39)

      call CONDEN( c(Ps402), r(Xs402), r(MRs402), c(Tds), c(Pds),
     >            r(Qcnd), r(IRRcnd), print )

      call PUMP  ( c(Ps402), c(Ps31), v(MRs31), c(EFFpmp), r(WRpmp),
     >    r(IRRpmp), print )

      call DELTAP( 'Air pollution dev 3 ', v(Pc211), c(dPap3),
     >    r(Pc212), print )

      call SPLITT( 'after fan #3        ', r(MRc12), r(MRc231),
     >    v(MRc25), print )

      call DELTAP( 'Air pollution dev 2 ', r(Pc111), -c(dPap2),
     >            r(Pc232), print)

C     ----- call these only when printed results needed -----
      if (print) then

        call FAN   ( 'FAN #1              ', c(Te61), r(Pe62), r(MRe61),
     >      r(ye61CD),r(ye61W),r(ye6102),r(ye61N2),r(ye61Ar),r(ye61CO),
     >      r(Pe63), c(EFFfn1), r(WRfn1), r(dPfn1), r(CFMfn1),
     >      r(IRRfn1), print )

        if(print) write(8,12) page,big,title

        call FAN   ( 'FAN #3              ', c(Tc21), r(Pc212),r(MRc12),
     >      r(yc11CD),r(yc11W),r(yc1102),r(yc11N2),r(yc11Ar),r(yc11CO),
     >      c(Pc22), c(EFFfn3), r(WRfn3), r(dPfn3), r(CFMfn3),
     >      r(IRRfn3), print )

        call FAN   ( 'FAN #4              ', c(Tc21), c(Pc22),r(MRc231),
     >      r(yc11CD),r(yc11W),r(yc1102),r(yc11N2),r(yc11Ar),r(yc11CO),
     >      r(Pc232), c(EFFfn4), r(WRfn4), r(dPfn4), r(CFMfn4),
     >      r(IRRfn4), print )
```

```
*       if(print) write(8,12) page,big,title

        call FAN   ( 'FAN #5              ', v(Tc26), r(Pc26), r(MRc26),
     >     r(yc11CD),r(yc11W),r(yc1102),r(yc11N2),r(yc11Ar),r(yc11CO),
     >     r(Pc27), c(EFFfn5), r(WRfn5), r(dPfn5), r(CFMfn5),
     >     r(IRRfn5), print )

        call GENBOX( 'Gas-Turbine         ', r(WRgtb), r(WRcmp),
     >     c(EFFgn1), c(EFFgb1), r(WRgn1), r(IRRgn1), print )

C       {--- note: GENBOX was designed for gas-turbine/compressor ---}
        call GENBOX( 'Steam-Turbine       ', r(WRstb1), -r(WRstb2),
     >     c(EFFgn2), c(EFFgb2), r(WRgn2), r(IRRgn2), print )

        call EFF   ( r(WRgn1), r(WRgn2), r(WRfn1), r(WRfn3), r(WRfn4),
     >               r(WRfn5), r(WRpmp), c(WRcnv), c(WRctr), r(WRdry),
     >               c(WRhml), c(WRscw), c(HHV), r(LHV1), r(LHV2),
     >               r(MRbdwd), c(Wb52), c(Wb51), c(HHVoil), c(LHVoil),
     >               r(MRoil), r(EFFHg), r(EFFHn),
     >               r(EFFL1g), r(EFFL3g), r(WRaux), r(WRgrs), print )

      endif
   12 format(1x,a1,9x,a1,a)
   14 format(10x,a1,a)
C(bug)RETURN
$list
      END

      subroutine GETR(rname,runits,numr)
$nolist
C-----------------------------------------------
C  GETR   -- gets the result names and units.
C-----------------------------------------------
      dimension Rname(*),Runits(*), R(200)
      character Rname*6,Runits*15,R*22
       R(  1) = 'Ablr  3square feet    '
       R(  2) = 'Ahx1  3square feet    '
       R(  3) = 'Ahx2  3square feet    '
       R(  4) = 'CFMfn13cu ft/min      '
       R(  5) = 'CFMfn33cu ft/min      '
       R(  6) = 'CFMfn43cu ft/min      '
       R(  7) = 'CFMfn53cu ft/min      '
       R(  8) = 'dPfn1 3in. h2o        '
       R(  9) = 'dPfn3 3in. h2o        '
       R( 10) = 'dPfn4 3in. h2o        '
       R( 11) = 'dPfn5 3in. h2o        '
       R( 12) = 'dPhx1g3in. h2o        '
       R( 13) = 'dPhx1w3in. h2o        '
       R( 14) = 'dPhx2a3in. h2o        '
       R( 15) = 'dPhx2g3in. h2o        '
       R( 16) = 'EFFHg 3% gross, HHV   '
       R( 17) = 'EFFHn 3% net, HHV     '
```

```
R( 18) = 'EFFL1g3% gross, LHV1  '
R( 19) = 'EFFL3g3% gross, LHV3  '
R( 20) = 'IRRblr3kW             '
R( 21) = 'IRRcmb3kW             '
R( 22) = 'IRRcmp3kW             '
R( 23) = 'IRRcnd3kW             '
R( 24) = 'IRRdry3kW             '
R( 25) = 'IRRfn13kW             '
R( 26) = 'IRRfn33kW             '
R( 27) = 'IRRfn43kW             '
R( 28) = 'IRRfn53kW             '
R( 29) = 'IRRgn13kW             '
R( 30) = 'IRRgn23kW             '
R( 31) = 'IRRgtb3kW             '
R( 32) = 'IRRhx13kW             '
R( 33) = 'IRRhx23kW             '
R( 34) = 'IRRinj3kW             '
R( 35) = 'IRRmxc3kW (mix ex cmb)'
R( 36) = 'IRRmxd3kW (mix in dry)'
R( 37) = 'IRRpmp3kW             '
R( 38) = 'IRRst13kW (1st stage) '
R( 39) = 'IRRst23kW (2nd stage) '
R( 40) = 'IRRtrm3kW             '
R( 41) = 'lendry3feet           '
R( 42) = 'LHV1  3BTU/lb -h2ocomb'
R( 43) = 'LHV2  3BTU/lb -h2owood'
R( 44) = 'Ma01  3lb/mol         '
R( 45) = 'Ma02  3lb/mol         '
R( 46) = 'Ma03  3lb/mol         '
R( 47) = 'MRa0213lb/s           '
R( 48) = 'MRa0223lb/s           '
R( 49) = 'MRa0323lb/s           '
R( 50) = 'MRa04 3lb/s           '
R( 51) = 'MRa06 3lb/s           '
R( 52) = 'MRb51 3lb/s           '
R( 53) = 'MRb52 3lb/s           '
R( 54) = 'MRbdwd3lb/s           '
R( 55) = 'MRc1113lb/s           '
R( 56) = 'MRc1123lb/s           '
R( 57) = 'MRc12 3lb/s           '
R( 58) = 'MRc2313lb/s           '
R( 59) = 'MRc26 3lb/s           '
R( 60) = 'MRe61 3lb/s           '
R( 61) = 'MRoil 3lb/s           '
R( 62) = 'MRs4023lb/s           '
R( 63) = 'Pa021 3psia           '
R( 64) = 'Pa022 3psia           '
R( 65) = 'Pa032 3psia           '
R( 66) = 'Pc111 3in. h2o        '
R( 67) = 'Pc112 3in. h2o        '
R( 68) = 'Pc113 3in. h2o        '
R( 69) = 'Pc13  3in. h2o        '
R( 70) = 'Pc14  3in. h2o        '
```

```
R( 71) = 'Pc212 3in. h2o       '
R( 72) = 'Pc232 3in. h2o       '
R( 73) = 'Pc26  3in. h2o       '
R( 74) = 'Pc27  3in. h2o       '
R( 75) = 'Pe62  3in. h2o       '
R( 76) = 'Pe63  3in. h2o       '
R( 77) = 'Ps38  3psia          '
R( 78) = 'Qcmb  3btu/s         '
R( 79) = 'Qcnd  3btu/s         '
R( 80) = 'Ta021 3F             '
R( 81) = 'Ta022 3F             '
R( 82) = 'Ta04  3F             '
R( 83) = 'Tc111 3F             '
R( 84) = 'Tc13  3F             '
R( 85) = 'Tc14  3F             '
R( 86) = 'Vhx1g 3ft/s          '
R( 87) = 'Vhx2g 3ft/s          '
R( 88) = 'WRaux 3kW            '
R( 89) = 'WRcmp 3kW            '
R( 90) = 'WRdry 3kW            '
R( 91) = 'WRfn1 3kW            '
R( 92) = 'WRfn3 3kW            '
R( 93) = 'WRfn4 3kW            '
R( 94) = 'WRfn5 3kW            '
R( 95) = 'WRgn1 3kW            '
R( 96) = 'WRgn2 3kW            '
R( 97) = 'WRgrs 3kW            '
R( 98) = 'WRgtb 3kW            '
R( 99) = 'WRpmp 3kW            '
R(100) = 'WRstb13kW (1st stage) '
R(101) = 'WRstb23kW (2nd stage) '
R(102) = 'Xs39  3lb/lb quality '
R(103) = 'Xs402 3lb/lb quality '
R(104) = 'ya01CD3mol fract CO2 '
R(105) = 'ya01W 3mol fract H2O '
R(106) = 'ya01023mol fract  O2 '
R(107) = 'ya01N23mol fract  N2 '
R(108) = 'ya01Ar3mol fract  Ar '
R(109) = 'ya01CO3mol fract  CO '
R(110) = 'ya02CD3mol fract CO2 '
R(111) = 'ya02W 3mol fract H2O '
R(112) = 'ya02023mol fract  O2 '
R(113) = 'ya02N23mol fract  N2 '
R(114) = 'ya02Ar3mol fract  Ar '
R(115) = 'ya02CO3mol fract  CO '
R(116) = 'ya03CD3mol fract CO2 '
R(117) = 'ya03W 3mol fract H2O '
R(118) = 'ya03023mol fract  O2 '
R(119) = 'ya03N23mol fract  N2 '
R(120) = 'ya03Ar3mol fract  Ar '
R(121) = 'ya03CO3mol fract  CO '
R(122) = 'ya04CD3mol fract CO2 '
R(123) = 'ya04W 3mol fract H2O '
```

```
      R(124) = 'ya04023mol fract  02  '
      R(125) = 'ya04N23mol fract  N2  '
      R(126) = 'ya04Ar3mol fract  Ar  '
      R(127) = 'ya04CO3mol fract  CO  '
      R(128) = 'yc11CD3mol fract CO2  '
      R(129) = 'yc11W 3mol fract H2O  '
      R(130) = 'yc11023mol fract  02  '
      R(131) = 'yc11N23mol fract  N2  '
      R(132) = 'yc11Ar3mol fract  Ar  '
      R(133) = 'yc11CO3mol fract  CO  '
      R(134) = 'ye61CD3mol fract CO2  '
      R(135) = 'ye61W 3mol fract H2O  '
      R(136) = 'ye61023mol fract  02  '
      R(137) = 'ye61N23mol fract  N2  '
      R(138) = 'ye61Ar3mol fract  Ar  '
      R(139) = 'ye61CO3mol fract  CO  '
     numr =139
     do 10 i = 1,numr
        read( R(i), 100 ) Rname(i), Runits(i)
  10 continue
     return
 100 format(a6,1x,a15)
     end
```

------ COMPRESSOR ------

Conditions at:	Inlet	Exit	
Pressure	14.70	132.42	
Temperature	59.00	595.52	F
Mass flow rate	36.68	36.68	lb/s
Exergy	.00	4644.47	kW

Inlet pressure loss	4.00	in. h2o
Bleeding air	4.00	%
Efficiency	83.20	%
Power to drive	5310.08	kW
Irreversibility	665.62	kW

```
------ TRIMBURNER ------
INPUT CONDITIONS TO TRIMBURNER
   Higher Heating Value             19870.0000 Btu/lb
   Fraction radiation loss               .0200
   Mass rate of inlet air              36.6792 lb/sec
   Inlet air humidity ratio              .0070
   Inlet air temperature             1450.0000 deg. F
   Exiting Flue-Gas Temperature      1612.0000
   Ambient air temperature             59.0000 deg. F
   Inlet air pressure                 129.3459 psia
INLET AND EXIT STREAMS
   Mass rate of fuel required                       =     .1004 lb/sec
   Mass rate of inlet air from turbine              =   36.6792 lb/sec
                                                        --------
   Total Inlet                                          36.7796
   Mass rate of flue-gas exiting                    =   36.7796 lb/sec
   Mass rate of ash exiting with flue-gas           =     .0000 lb/sec
                                                        --------
   Total Exit                                           36.7796

FLUE-GAS ANALYSIS
    Constituent          Mole Fraction
   Carbon dioxide          .0059
   Nitrogen                .7703
   Oxygen                  .1985
   Water vapor             .0161
   Carbon monoxide         .0001
   Argon                   .0092
   Molecular weight of flue-gas                        =  28.8506 lbs/mol
   Humidity ratio (lb H2O/lb dry gas) of flue-gas      =      .0101
   Excess air level                                    =  2426.8800 percent
   Air-Fuel Ratio                                      =   365.5
   Exit pressure                                       =   128.3459 psia
ENERGY BALANCE
   Source or Sink                          Btu/lb fuel         Btu/sec

   Heat released by combustion               19870.0          1994.0760
   Heat provided by inlet air               128860.0         12931.8900
                                            --------         ----------
   Total                                    148730.0         14925.9600
   Heat loss from radiation                    397.4            39.8815
   Heat loss from formation of CO              197.9            19.8621
   Heat absorbed by exhaust gases           148134.8         14866.2400
                                            --------         ----------
   Total                                    148730.2         14925.9800
SECOND LAW ANALYSIS
   Irreversibility rate  =    731.3  KW
   Second Law Efficiency =     92.49 Percent
```

------ GAS TURBINE ------

Conditions at:	Inlet	Exit
Pressure	128.35	26.00
Temperature	1612.00	823.41 F
Mass flow rate	36.78	36.78 lb/s
Exergy	11617.67	3105.24 kW

Mass flow rate	36.78 lb/s
Cooling flow	1.53 lb/s
Power output	8242.94 kW
Irreversibility	463.06 kW

------ COMBUSTOR ------

Air-Fuel ratio (dry to dry)	17.0000
Fraction radiation loss	.0200
Fraction unburned carbon	.0200
Fraction of carbon burned to CO	.0100

Excess air level = 154.4423 percent

Adiabatic flame temperature = 1845.2 deg F

Irreversibility rate = 9491.9 KW

Second Law Efficiency = 60.15 Percent

------ MIXER after combustor ------

Irreversibility = 337.39 kW

------ Cyclone #3 ------

Pressure loss = 5.000

------ Splitter: after combustor ------

------ STEAM-GAS HEAT EXCHANGER ------

Flow conditions:

		flue gas	steam
Temperature at inlet, F	:	1675.000	700.891
Pressure at inlet, psia	:	15.057	640.000
Massflow, lbm/sec :	:	48.676	7.542
Temperature at exit, F	:	1622.001	900.000
Pressure at exit, psia	:	15.053	633.585
Cp - values, btu/lbm F	:	.310	.533
Viscosity, lbm/s ft	:	.30799E-04	.17247E-04
Conductivity,btu/h ft R	:	.43941E-01	.36074E-01
Exergy at inlet, KW	:	14206.020	4181.206
Exergy at exit, KW	:	13569.220	4672.684

Specifications of Heat Exchanger:

Tube arrangement:	in-line	
Spacing of tubes normal to flow, Sn/d:	2.000	
Spacing of tubes parallel to flow, Sp/d:	2.000	
Number of tubes normal to flow:	60.000	
Number of tubes parallel to flow:	1.000	
Number of U's:	1.000	
Outside diameter of pipes:	.750	inches
Inside diameter of pipes	.560	inches
Thermal conductivity of the tubes:	20.000	btu/h/ft/F
Rougness factor inside the tubes (k/D):	.002	
Length of the tubes:	9.359	ft
Height of the heat exchanger:	9.484	ft
Width of the heat exchanger:	7.500	ft
Total heat transfer area:	220.513	sqft
Maximum velocity:	62.905	ft/sec

Heat Transfer Data:

"C" - to calculate Nusselt Number:	.254	
"X" - to calculate Nusselt Number:	.632	
Safety factor on the shell side:	.900	
Correction factor due to cross flow:	.999	
Correction factor due to few pipes:	.797	
H out - heat transfer coefficient:	18.299	btu/h/sqft/F
H in - heat transfer coefficient:	292.727	btu/h/sqft/F
Fouling in heat exchanger:	.005	
U - Overall heat transfer coefficient	15.461	btuh/sqft/F
UA - value:	3409.280	btu/h/F
Log-Mean-Temperature-Difference :	845.953	F
Maximum wall temperature:	945.596	F
Average wall temperature:	850.340	F
Total heat transfer:	843.966	kW
Irreversibility:	145.322	KW

Pressure drops:

Pressure drop inside pipes	:	6.415	psi
Pressure drop pr. ft of pipe	:	.343	psi/ft
Pressure drop on gas side	:	.107	in W.C.

------ AIR-GAS HEAT EXCHANGER ------

Flow conditions:

		flue gas	air
Temperature at inlet, F	:	1622.001	595.520
Pressure at inlet, psia	:	15.053	132.418
Massflow, lbm/sec :	:	48.676	36.679
Temperature at exit, F	:	1053.085	1450.000
Pressure at exit, psia	:	14.927	129.346
Cp - values, btu/lbm F	:	.301	.266
Viscosity, lbm/s ft	:	.27820E-04	.24998E-04
Conductivity,btu/h ft R	:	.39141E-01	.33031E-01
Exergy at inlet, KW	:	13569.220	4644.467
Exergy at exit, KW	:	7316.601	10252.590

Specifications of Heat Exchanger:

Tube arrangement:	staggered	
Spacing of tubes normal to flow, Sn/d:	1.700	
Spacing of tubes parallel to flow, Sp/d:	1.500	
Number of tubes normal to flow:	72.000	
Number of tubes parallel to flow:	8.000	
Number of U's:	3.000	
Outside diameter of pipes:	1.000	inches
Inside diameter of pipes	.810	inches
Thermal conductivity of the tubes:	20.000	btu/h/ft/F
Rougness factor inside the tubes (k/D):	.002	
Length of the tubes:	9.020	ft
Height of the heat exchanger:	9.145	ft
Width of the heat exchanger:	10.200	ft
Total heat transfer area:	8160.936	sqft
Maximum velocity:	49.527	ft/sec

Heat Transfer Data:

"C" - to calculate Nusselt Number:	.507	
"X" - to calculate Nusselt Number:	.564	
Safety factor on the shell side:	.900	
Correction factor due to cross flow:	1.000	
Correction factor due to few pipes:	1.000	
H out - heat transfer coefficient:	20.351	btu/h/sqft/F
H in - heat transfer coefficient:	49.754	btu/h/sqft/F
Fouling in heat exchanger:	.005	
U - Overall heat transfer coefficient	12.596	btuh/sqft/F
UA - value:	102794.100	btu/h/F
Log-Mean-Temperature-Difference :	291.863	F
Maximum wall temperature:	1499.930	F
Average wall temperature:	1114.138	F
Total heat transfer:	8791.349	kW
Irreversibility:	644.493	KW

Pressure drops:

Pressure drop inside pipes	:	3.073	psi
Pressure drop pr. ft of pipe	:	.057	psi/ft
Pressure drop on gas side	:	3.512	in W.C.

------ MIXER to dryer ------
Irreversibility = .00 kW

------ Air pollution dev 1 ------
Pressure loss = .000

------ Cyclones #1, #2 ------
Pressure loss = .000

------ Bag house ------
Pressure loss = .000

------ WASTE HEAT BOILER ------

Flow conditions:		flue gas	steam
Temperature at inlet, F	:	1053.085	115.223
Pressure at inlet, psia	:	14.927	650.000
Massflow, lbm/sec :	:	48.676	7.542
Temperature at exit, F	:	350.000	700.891
Pressure at exit, psia	:	14.637	640.000

Heat Transfer Data:		
dT at pinch point:	85.102	F
Saturation temperature:	494.721	F
Superheater surface area:	920.261	sqft
Boiler surface area:	8690.653	sqft
Economizer surface area:	6089.179	sqft
Total heat transfer area:	15700.090	sqft
Total heat transfer:	10015.990	KW
Irreversibility:	1301.889	KW

Pressure drops:			
Pressure drop on steam side (approx)	:	10.000	psi
Pressure drop on gas side	:	8.000	in W.C.

```
------ STEAM TURBINE ------
Conditions at:              Inlet         Exit
   Pressure                633.59         1.50
   Temperature             900.00       115.22 F
   Mass flow rate            7.54         7.54 lb/s
   Exergy                 4517.63       620.08 kW
Efficiency        80.000 %
Exit quality      93.727 %
Power output    3180.242 kW
Irrevers.        717.305 kW

------ CONDENSER  ------
Pressure                    1.50 psia
Quality of steam           93.73 %
Mass flow rate              7.54 lbm/s
Cooling load             7679.78 KW
Coolant flow rate         727.80 lbm/s
Irreversibility           751.06 KW

------ PUMP ------
Mass flow rate          7.542 lbm/s
Inlet pressure          1.500 psia
Exit pressure         650.000 psia
Efficiency             68.000 %
Power required         23.779 kW
Irreversibility         7.609 kW

------ Air pollution dev 3  ------
Pressure loss =       5.000

------ Splitter: after fan #3        ------

------ Air pollution dev 2  ------
Pressure loss =        .000

------ Fan: FAN #1                    ------
Conditions at:              Inlet         Exit
   Pressure                   .70          .70
   Temperature             350.00       350.00 F
   Mass flow rate           42.73        42.73 lb/s
   Exergy                     .00          .00 kW
Power required           .00 kW
Irreversibility          .00 kW
Delta P                  .00 in. h2o
Flow rate           53658.11 cu ft/min
```

```
------ Fan: FAN #3                        ------
Conditions at:               Inlet         Exit
   Pressure                  -6.62          .70
   Temperature              350.00       350.00 F
   Mass flow rate            48.68        48.68 lb/s
   Exergy                      .00          .00 kW
Power required       84.25 kW
Irreversibility      31.17 kW
Delta P               7.32 in. h2o
Flow rate         61677.91 cu ft/min

------ Fan: FAN #4                        ------
Conditions at:               Inlet         Exit
   Pressure                    .70        15.00
   Temperature              350.00       350.00 F
   Mass flow rate             5.95         5.95 lb/s
   Exergy                      .00          .00 kW
Power required       19.58 kW
Irreversibility       7.25 kW
Delta P              14.30 in. h2o
Flow rate          7337.09 cu ft/min

------ Fan: FAN #5                        ------
Conditions at:               Inlet         Exit
   Pressure                    .70          .70
   Temperature              350.00       350.00 F
   Mass flow rate            42.73        42.73 lb/s
   Exergy                      .00          .00 kW
Power required         .00 kW
Irreversibility        .00 kW
Delta P                .00 in. h2o
Flow rate         53658.11 cu ft/min

------ Generator/Gear box: Gas-Turbine           ------
Irreversibility        244.01 kW

------ Generator/Gear box: Steam-Turbine         ------
Irreversibility        264.60 kW

------ Plant Summary -----
Auxilliary Power (kW)            Generator Power (kW)
  Pump              23.78          Gasturb/Compressor       2688.85
  Cooling twr       45.00          Steam turbine            2915.65
  Fan #1              .00
  Fan #3            84.25        First Law Summary (kW)
  Fan #4            19.58          Work Rate (gross)     5604.49
  Fan #5              .00          Work Rate (net)       5409.88
  Conveyor          12.00          Fuel Energy (HHV)    22795.46
  Hammermill          .00
  Screw             10.00
  Dryer               .00
  TOTAL......      194.61
Plant Efficiency (%)          HHV       LHV1       LHV2       LHV3
  Gross                      24.6       26.4       29.9       29.9
  Net                        23.7       25.4       28.8       28.8
```

5 iterations, specified tolerance: .0100, actual tolerance: .0001

----- Constants -----

AFrat	17.0000	lb air/lb fuel	Pe64	.7000	in. h2o
chx1	.2540		Prcinj	.0000	% injection
chx2	.5070		Prcmp	9.1000	press. ratio
cmpbld	4.0000	%	Ps31	650.0000	psia
dPap1	.0000	in. h2o	Ps39	1.5000	psia
dPap2	.0000	in. h2o	Ps402	1.5000	psia
dPap3	5.0000	in. h2o	radlos	.0200	frac. energy in
dPbag	.0000	in. h2o	radtrm	.0200	frac. HHVoil
dPblrg	8.0000	in. h2o	rufhx1	.0020	
dPblrw	10.0000	psia	rufhx2	.0020	
dPcmb	11.0000	in. h2o	SnDhx1	2.0000	
dPcmp	4.0000	in. h2o	SnDhx2	1.7000	
dPcyc1	.0000	in. h2o	SpDhx1	2.0000	
dPcyc3	5.0000	in. h2o	SpDhx2	1.5000	
dPdry	.0000	in. h2o	Ta01	59.0000	F
dPgtb	2.0000	in. h2o	Ta031	1450.0000	F
dPtrim	1.0000	psi	Ta032	1612.0000	F
drylos	.0500	fract ht loss	Tc112	1675.0000	F
EFFcmp	83.2000	%	Tc21	350.0000	F
EFFfn1	63.0000	%	Tds	59.0000	F
EFFfn3	63.0000	%	Te61	350.0000	F
EFFfn4	63.0000	%	Ts38	900.0000	F
EFFfn5	63.0000	%	Tsuper	207.0000	F
EFFgb1	96.0000	%	typhx1	1.0000	1 = in-line
EFFgb2	96.0000	%	typhx2	2.0000	2 = staggered
EFFgn1	95.5000	%	Ublr	10.0000	btu/hr-ft2-F
EFFgn2	95.5000	%	Uecon	12.0000	btu/hr-ft2-F
EFFgtb	91.2000	%	unbcbn	.0200	lb/lb
EFFpmp	68.0000	%	Usuper	8.0000	btu/hr-ft2-F
EFFstb	80.0000	%	Wa01	.0070	lb h2o/lb dry a
fouhx1	.0050	fouling factor	Wb51	.5000	lb h2o/lb fuel
fouhx2	.0050	fouling factor	Wb52	.5000	lb h2o/lb fuel
fshx1	.9000	factor safety	WRcnv	12.0000	kW
fshx2	.9000	factor safety	WRctr	45.0000	kW
HHV	8800.0000	btu/lb	WRhml	.0000	kW
HHVoil	19870.0000	btu/lb	WRscw	10.0000	kW
IDhx1	.5600	inches	Xash	.0050	lb ash/lb dry f
IDhx2	.8100	inches	Xashoi	.0000	lb ash/lb oil
Kphx1	20.0000	btu/hr-ft-F	XC	.5230	lb C/lb dry fue
Kphx2	20.0000	btu/hr-ft-F	XCCO	.0100	lb C burned CO/
LHVoil	18660.0000	btu/lb	XCCOoi	.0100	lb C burned CO/
MRc16m	99999.0000	lb/s	XCoil	.8700	lb C/lb oil
MRc23m	1.0000	lb/s	Xdirt	.0350	lb dirt/lb dry
MRstp	38.7500	lb/s	XH2	.0630	lb H2/lb dry fu
npnhx1	60.0000	pipes normal	XH2oil	.1270	lb H2/lb oil
npnhx2	72.0000	pipes normal	xhx1	.6320	
npphx1	1.0000	pipes parallel	xhx2	.5640	
npphx2	8.0000	pipes parallel	XN2	.0010	lb N2/lb dry fu
nuhx1	1.0000	no. of U's	XN2oil	.0015	lb N2/lb oil
nuhx2	3.0000	no. of U's	XO2	.4050	lb O2/lb dry fu
ODhx1	.7500	inches	XO2oil	.0015	lb O2/lb oil
ODhx2	1.0000	inches	ydsCD	.0003	
Pa01	14.6960	psia	ydsW	.0111	
Pa04	26.0000	in. h2o	ydsO2	.2072	
Pc22	.7000	in. h2o	ydsN2	.7722	
Pds	14.6960	psia	ydsAr	.0092	
Pe61	.7000	in. h2o	ydsCO	.0000	

----- Variables -----

MRc16	.0000 lb/s	Ps36	640.0000 psia
MRc25	42.7308 lb/s	Tc26	350.0000 F
MRs31	7.5418 lb/s	Ts36	700.8911 F
Pa031	129.3458 psia	Ts39	115.2230 F
Pc211	-1.6193 in. h2o	Ts402	115.2230 F
Pc29	.7000 in. h2o		

----- Int.Calcs -----

Ablr	15700.0900	square feet	Pc212	-6.6193	in. h2o
Ahx1	220.5134	square feet	Pc232	15.0000	in. h2o
Ahx2	8160.9360	square feet	Pc26	.7000	in. h2o
CFMfn1	53658.1100	cu ft/min	Pc27	.7000	in. h2o
CFMfn3	61677.9100	cu ft/min	Pe62	.7000	in. h2o
CFMfn4	7337.0860	cu ft/min	Pe63	.7000	in. h2o
CFMfn5	53658.1100	cu ft/min	Ps38	633.5854	psia
dPfn1	.0000	in. h2o	Qcmb	19633.4900	btu/s
dPfn3	7.3193	in. h2o	Qcnd	7278.0350	btu/s
dPfn4	14.3000	in. h2o	Ta021	595.5196	F
dPfn5	.0000	in. h2o	Ta022	595.5196	F
dPhx1g	.1070	in. h2o	Ta04	823.4147	F
dPhx1w	6.4146	in. h2o	Tc111	1845.1520	F
dPhx2a	3.0726	in. h2o	Tc13	1622.0010	F
dPhx2g	3.5123	in. h2o	Tc14	1053.0850	F
EFFHg	24.5860	% gross, HHV	Vhx1g	62.9051	ft/s
EFFHn	23.7323	% net, HHV	Vhx2g	49.5270	ft/s
EFFL1g	26.3566	% gross, LHV1	WRaux	194.6123	kW
EFFL3g	29.8604	% gross, LHV3	WRcmp	5310.0850	kW
IRRblr	1301.8890	kW	WRdry	.0000	kW
IRRcmb	9491.8920	kW	WRfn1	.0000	kW
IRRcmp	665.6182	kW	WRfn3	84.2519	kW
IRRcnd	751.0623	kW	WRfn4	19.5813	kW
IRRdry	.0000	kW	WRfn5	.0000	kW
IRRfn1	.0000	kW	WRgn1	2688.8450	kW
IRRfn3	31.1732	kW	WRgn2	2915.6460	kW
IRRfn4	7.2451	kW	WRgrs	5604.4910	kW
IRRfn5	.0000	kW	WRgtb	8242.9440	kW
IRRgn1	244.0139	kW	WRpmp	23.7791	kW
IRRgn2	264.5962	kW	WRstb1	3180.2420	kW (1st stage)
IRRgtb	463.0634	kW	WRstb2	.0000	kW (2nd stage)
IRRhx1	145.3223	kW	Xs39	.9373	lb/lb quality
IRRhx2	644.4932	kW	Xs402	.9373	lb/lb quality
IRRinj	.0000	kW	ya01CD	.0003	mol fract CO2
IRRmxc	337.3940	kW (mix ex cmb)	ya01W	.0111	mol fract H2O
IRRmxd	.0000	kW (mix in dry)	ya0102	.2072	mol fract O2
IRRpmp	7.6093	kW	ya01N2	.7722	mol fract N2
IRRst1	717.3051	kW (1st stage)	ya01Ar	.0092	mol fract Ar
IRRst2	.0000	kW (2nd stage)	ya01CO	.0000	mol fract CO
IRRtrm	731.2554	kW	ya02CD	.0003	mol fract CO2
lendry	.0000	feet	ya02W	.0111	mol fract H2O
LHV1	8203.2200	BTU/lb -h2ocomb	ya0202	.2072	mol fract O2
LHV2	7142.1630	BTU/lb -h2owood	ya02N2	.7722	mol fract N2
Ma01	28.8445	lb/mol	ya02Ar	.0092	mol fract Ar
Ma02	28.8445	lb/mol	ya02CO	.0000	mol fract CO
Ma03	28.8506	lb/mol	ya03CD	.0059	mol fract CO2
MRa021	36.6792	lb/s	ya03W	.0161	mol fract H2O
MRa022	36.6792	lb/s	ya0302	.1985	mol fract O2
MRa032	36.7796	lb/s	ya03N2	.7703	mol fract N2
MRa04	38.3079	lb/s	ya03Ar	.0092	mol fract Ar
MRa06	1.5283	lb/s	ya03CO	.0001	mol fract CO
MRb51	4.5402	lb/s	ya04CD	.0057	mol fract CO2
MRb52	4.5402	lb/s	ya04W	.0159	mol fract H2O
MRbdwd	2.2311	lb/s	ya0402	.1988	mol fract O2
MRc111	42.7308	lb/s	ya04N2	.7703	mol fract N2
MRc112	48.6762	lb/s	ya04Ar	.0092	mol fract Ar
MRc12	48.6762	lb/s	ya04CO	.0001	mol fract CO
MRc231	5.9454	lb/s	yc11CD	.0672	mol fract CO2
MRc26	42.7308	lb/s	yc11W	.1416	mol fract H2O
MRe61	42.7308	lb/s	yc1102	.1074	mol fract O2
MRoil	.1004	lb/s	yc11N2	.6751	mol fract N2
MRs402	7.5418	lb/s	yc11Ar	.0080	mol fract Ar

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Pa021	132.4185	psia	yc11CO	.0007	mol fract CO
Pa022	132.4185	psia	ye61CD	.0672	mol fract CO2
Pa032	128.3459	psia	ye61W	.1416	mol fract H2O
Pc111	15.0000	in. h2o	ye61O2	.1074	mol fract O2
Pc112	15.0000	in. h2o	ye61N2	.6751	mol fract N2
Pc113	10.0000	in. h2o	ye61Ar	.0080	mol fract Ar
Pc13	9.8930	in. h2o	ye61CO	.0007	mol fract CO
Pc14	6.3807	in. h2o			

APPENDIX C

Results of Simulations

Systems 1, 2, 3-2, 4, 5-1, 5-2

Parametric Study of System 5-2

Name	Units	System Number 1	2	3-2	4	5-1	5-2	Percent Change from Sys 2: 1	3-2	4	5-1	5-2
AFrat	lb air/lb fuel	17	17	17	17	17	17	0%	0%	0%	0%	0%
Ablr	square feet	15895	10967	11835	11656	15218	15700	45%	8%	6%	39%	43%
Ahx1	square feet	223.18	204.93	191.08	233.42	190.2	220.51	9%	-7%	14%	-7%	8%
Ahx2	square feet	7946.5	6933.3	8447.3	6517.1	8929	8160.9	15%	22%	-6%	29%	18%
CFMfn1	cu ft/min	42980	43000	45895	43092	53544	53658	0%	7%	0%	25%	25%
CFMfn3	cu ft/min	62751	76949	74789	80915	58327	61678	-18%	-3%	5%	-24%	-20%
CFMfn4	cu ft/min	16259	19867	15229	23540	4242.2	7337.1	-18%	-23%	18%	-79%	-63%
CFMfn5	cu ft/min	54908	55259	57990	55305	53544	53658	-1%	5%	0%	-3%	-3%
EFFHg	% gross, HHV	24.454	24.477	24.708	26.271	22.626	24.586	0%	1%	7%	-8%	0%
EFFHn	% net, HHV	22.065	21.796	22.205	23.658	21.802	23.732	1%	2%	9%	0%	9%
EFFL1g	% gross, LHV1	26.233	26.258	26.506	28.163	24.272	26.357	0%	1%	7%	-8%	0%
EFFL3g	% gross, LHV3	28.712	28.739	29.01	30.557	27.877	29.86	0%	1%	6%	-3%	4%
EFFcmp	%	83.2	83.2	83.2	83.2	83.2	83.2	0%	0%	0%	0%	0%
EFFfn1	%	63	63	63	63	63	63	0%	0%	0%	0%	0%
EFFfn3	%	63	63	63	63	63	63	0%	0%	0%	0%	0%
EFFfn4	%	63	63	63	63	63	63	0%	0%	0%	0%	0%
EFFfn5	%	63	63	63	63	63	63	0%	0%	0%	0%	0%
EFFgb1	%	96	96	96	96	96	96	0%	0%	0%	0%	0%
EFFgb2	%	96	96	96	96	96	96	0%	0%	0%	0%	0%
EFFgn1	%	95.5	95.5	95.5	95.5	95.5	95.5	0%	0%	0%	0%	0%
EFFgn2	%	95.5	95.5	95.5	95.5	95.5	95.5	0%	0%	0%	0%	0%
EFFgtb	%	91.2	91.2	91.2	91.2	91.2	91.2	0%	0%	0%	0%	0%
EFFpmp	%	68	68	68	68	68	68	0%	0%	0%	0%	0%
EFFstb	%	80	80	80	80	80	80	0%	0%	0%	0%	0%
HHV	btu/lb	8800	8800	8800	8800	8800	8800	0%	0%	0%	0%	0%
HHVoil	btu/lb	19870	19870	19870	19870	19870	19870	0%	0%	0%	0%	0%
IDhx1	inches	0.56	0.56	0.56	0.56	0.56	0.56	0%	0%	0%	0%	0%
IDhx2	inches	0.81	0.81	0.81	0.81	0.81	0.81	0%	0%	0%	0%	0%
IRRblr	kW	1346.6	1664.2	1483.3	1869.8	1119.6	1301.9	-19%	-11%	12%	-33%	-22%
IRRcmb	kW	7845	7846.8	7975.7	7591.6	9724	9491.9	0%	2%	-3%	24%	21%
IRRcmp	kW	665.62	665.62	665.62	665.62	665.62	665.62	0%	0%	0%	0%	0%
IRRcnd	kW	769.84	769.8	549.01	847.08	674.49	751.06	0%	-29%	10%	-12%	-2%
IRRdry	kW	623.2	623.06	545.89	616.7	0	0	0%	-12%	-1%	-100%	-100%
IRRfn1	kW	16.917	16.925	18.064	16.961	0	0	0%	7%	0%	-100%	-100%
IRRfn3	kW	32.904	77.413	65.317	89.024	26.195	31.173	-57%	-16%	15%	-66%	-60%
IRRfn4	kW	16.28	13.033	9.9906	15.443	4.189	7.2451	25%	-23%	18%	-68%	-44%
IRRfn5	kW	32.228	13.355	14.015	13.366	0	0	141%	5%	0%	-100%	-100%
IRRgn1	kW	187.96	188.63	240.13	245.21	187.32	244.01	0%	27%	30%	-1%	29%
IRRgn2	kW	271.44	271.21	224.04	298.27	237.73	264.6	0%	-17%	10%	-12%	-2%
IRRgtb	kW	462.37	462.95	490.14	463.99	461.8	463.06	0%	6%	0%	0%	0%
IRRhx1	kW	149.24	149.25	142.64	165.9	129.47	145.32	0%	-4%	11%	-13%	-3%
IRRhx2	kW	650.64	695.12	729.67	720.03	619.12	644.49	-6%	5%	4%	-11%	-7%
IRRinj	kW	0	0	358.83	0	0	0	ERR	ERR	ERR	ERR	ERR
IRRmxc	kW (mix ex cmb)	808.12	712.65	564.73	871.53	189.34	337.39	13%	-21%	22%	-73%	-53%
IRRmxd	kW (mix in dry)	260.79	0	0	0	0	0	ERR	ERR	ERR	ERR	ERR
IRRpmp	kW	7.8067	7.7993	7.4757	8.5803	6.835	7.6093	0%	-4%	10%	-12%	-2%
IRRst1	kW (1st stage)	735.87	735.24	149.08	808.59	644.49	717.31	0%	-80%	10%	-12%	-2%
IRRst2	kW (2nd stage)	0	0	361.37	0	0	0	ERR	ERR	ERR	ERR	ERR
IRRtrm	kW	0	0	0	731.25	0	731.26	ERR	ERR	ERR	ERR	ERR
Kphx1	btu/hr-ft-F	20	20	20	20	20	20	0%	0%	0%	0%	0%

Name	Units	1	2	3-2	4	5-1	5-2	1	3-2	4	5-1	5-2
		System Number						Percent Change from Sys 2				
Kphx2	btu/hr-ft-F	20	20	20	20	20	20	0%	0%	0%	0%	0%
LHV1	BTU/lb -h2ocomb	8203.2	8203.2	8203.2	8203.2	8203.2	8203.2	0%	0%	0%	0%	0%
LHV2	BTU/lb -h2owood	8033.2	8033.2	8033.2	8033.2	7142.2	7142.2	0%	0%	0%	-11%	-11%
LHVoil	btu/lb	18660	18660	18660	18660	18660	18660	0%	0%	0%	0%	0%
MRa021	lb/s	36.679	36.679	36.679	36.679	36.679	36.679	0%	0%	0%	0%	0%
MRa022	lb/s	36.679	36.679	38.513	36.679	36.679	36.679	0%	5%	0%	0%	0%
MRa032	lb/s	36.679	36.679	38.513	36.78	36.679	36.78	0%	5%	0%	0%	0%
MRa04	lb/s	38.208	38.208	40.041	38.308	38.208	38.308	0%	5%	0%	0%	0%
MRa06	lb/s	1.5283	1.5283	1.5283	1.5283	1.5283	1.5283	0%	0%	0%	0%	0%
MRb51	lb/s	4.5419	4.5419	4.5419	4.5402	4.5419	4.5402	0%	0%	0%	0%	0%
MRb52	lb/s	2.6143	2.6143	2.6143	2.6134	4.5419	4.5402	0%	0%	0%	74%	74%
MRbdwd	lb/s	2.2319	2.2319	2.2319	2.2311	2.2319	2.2311	0%	0%	0%	0%	0%
MRc111	lb/s	40.702	40.702	42.536	40.804	42.63	42.731	0%	5%	0%	5%	5%
MRc112	lb/s	54.396	55.442	53.788	58.298	46.067	48.676	-2%	-3%	5%	-17%	-12%
MRc12	lb/s	51.459	55.442	53.788	58.298	46.067	48.676	-7%	-3%	5%	-17%	-12%
MRc16	lb/s	2.9384	0	0	0	0	0	ERR	ERR	ERR	ERR	ERR
MRc16m	lb/s	1	99999	99999	99999	99999	99999	-100%	0%	0%	0%	0%
MRc231	lb/s	13.694	14.739	11.252	17.494	3.4367	5.9454	-7%	-24%	19%	-77%	-60%
MRc23m	lb/s	1	1	1	1	1	1	0%	0%	0%	0%	0%
MRc25	lb/s	37.765	40.702	42.536	40.804	42.63	42.731	-7%	5%	0%	5%	5%
MRc26	lb/s	40.682	40.702	42.536	40.804	42.63	42.731	0%	5%	0%	5%	5%
MRe61	lb/s	42.609	42.63	44.464	42.731	42.63	42.731	0%	4%	0%	0%	0%
MRoil	lb/s	0	0	0	0.10036	0	0.10036	ERR	ERR	ERR	ERR	ERR
MRs31	lb/s	7.7374	7.7301	7.4094	8.5041	6.7743	7.5418	0%	-4%	10%	-12%	-2%
MRs402	lb/s	7.7301	7.7301	5.5754	8.5041	6.7743	7.5418	0%	-28%	10%	-12%	-2%
MRstp	lb/s	38.75	38.75	38.75	38.75	38.75	38.75	0%	0%	0%	0%	0%
Ma01	lb/mol	28.845	28.845	28.845	28.845	28.845	28.845	0%	0%	0%	0%	0%
Ma02	lb/mol	28.845	28.845	28.042	28.845	28.845	28.845	0%	-3%	0%	0%	0%
Ma03	lb/mol	28.845	28.845	28.042	28.851	28.845	28.851	0%	-3%	0%	0%	0%
ODhx1	inches	0.75	0.75	0.75	0.75	0.75	0.75	0%	0%	0%	0%	0%
ODhx2	inches	1	1	1	1	1	1	0%	0%	0%	0%	0%
Pa01	psia	14.696	14.696	14.696	14.696	14.696	14.696	0%	0%	0%	0%	0%
Pa021	psia	132.42	132.42	132.42	132.42	132.42	132.42	0%	0%	0%	0%	0%
Pa022	psia	132.42	132.42	132.42	132.42	132.42	132.42	0%	0%	0%	0%	0%
Pa031	psia	129.43	129.81	128.83	129.97	129.05	129.35	0%	-1%	0%	-1%	0%
Pa032	psia	129.42	129.81	128.83	128.97	129.05	128.35	0%	-1%	-1%	-1%	-1%
Pa04	in. h2o	26	26	26	26	26	26	0%	0%	0%	0%	0%
Pc111	in. h2o	15	15	15	15	15	15	0%	0%	0%	0%	0%
Pc112	in. h2o	15	15	15	15	15	15	0%	0%	0%	0%	0%
Pc113	in. h2o	10	10	10	10	10	10	0%	0%	0%	0%	0%
Pc13	in. h2o	9.887	9.8485	9.8332	9.8687	9.8737	9.893	0%	0%	0%	0%	0%
Pc14	in. h2o	5.9064	3.931	5.8525	2.5672	7.1961	6.3807	50%	49%	-35%	83%	62%
Pc211	in. h2o	-2.0936	-4.069	-2.1475	-5.4328	-0.8038	-1.6193	-49%	-47%	34%	-80%	-60%
Pc212	in. h2o	-2.0936	-9.069	-7.1475	-10.433	-5.8039	-6.6193	-77%	-21%	15%	-36%	-27%
Pc22	in. h2o	5.5	5.5	5.5	5.5	0.7	0.7	0%	0%	0%	-87%	-87%
Pc232	in. h2o	20	15	15	15	15	15	33%	0%	0%	0%	0%
Pc26	in. h2o	5.5	5.5	5.5	5.5	0.7	0.7	0%	0%	0%	-87%	-87%
Pc27	in. h2o	14	9	9	9	0.7	0.7	56%	0%	0%	-92%	-92%
Pc29	in. h2o	9	9	9	9	0.7	0.7	0%	0%	0%	-92%	-92%
Pds	psia	14.696	14.696	14.696	14.696	14.696	14.696	0%	0%	0%	0%	0%

Name	Units	System Number 1	2	3-2	4	5-1	5-2	Percent Change from Sys 2: 1	3-2	4	5-1	5-2
Pe61	in. h2o	0	0	0	0	0.7	0.7	ERR	ERR	ERR	ERR	ERR
Pe62	in. h2o	-2.5	-2.5	-2.5	-2.5	0.7	0.7	0%	0%	0%	-128%	-128%
Pe63	in. h2o	3.2	3.2	3.2	3.2	0.7	0.7	0%	0%	0%	-78%	-78%
Pe64	in. h2o	0.7	0.7	0.7	0.7	0.7	0.7	0%	0%	0%	0%	0%
Prcinj	% injection	0	0	5	0	0	0	ERR	ERR	ERR	ERR	ERR
Prcmp	press. ratio	9.1	9.1	9.1	9.1	9.1	9.1	0%	0%	0%	0%	0%
Ps31	psia	650	650	650	650	650	650	0%	0%	0%	0%	0%
Ps36	psia	640	640	640	640	640	640	0%	0%	0%	0%	0%
Ps38	psia	633.18	633.74	634.64	631.39	635.53	633.59	0%	0%	0%	0%	0%
Ps39	psia	1.5	1.5	150	1.5	1.5	1.5	0%	9900%	0%	0%	0%
Ps402	psia	1.5	1.5	1.5	1.5	1.5	1.5	0%	0%	0%	0%	0%
Qcmb	btu/s	19641	19641	19641	19633	19641	19633	0%	0%	0%	0%	0%
Qcnd	btu/s	7460	7459.6	5320.1	8208.5	6536	7278	0%	-29%	10%	-12%	-2%
SnDhx1		2	2	2	2	2	2	0%	0%	0%	0%	0%
SnDhx2		1.7	1.7	1.7	1.7	1.7	1.7	0%	0%	0%	0%	0%
SpDhx1		2	2	2	2	2	2	0%	0%	0%	0%	0%
SpDhx2		1.5	1.5	1.5	1.5	1.5	1.5	0%	0%	0%	0%	0%
Ta01	F	59	59	59	59	59	59	0%	0%	0%	0%	0%
Ta021	F	595.52	595.52	595.52	595.52	595.52	595.52	0%	0%	0%	0%	0%
Ta022	F	595.52	595.52	592.33	595.52	595.52	595.52	0%	-1%	0%	0%	0%
Ta031	F	1450	1450	1450	1450	1450	1450	0%	0%	0%	0%	0%
Ta032	F	1450	1450	1450	1612	1450	1612	0%	0%	11%	0%	11%
Ta04	F	714.65	713.87	725.22	822.05	715.39	823.41	0%	2%	15%	0%	15%
Tc111	F	2084.8	2084.2	1978.3	2158.6	1773.9	1845.2	0%	-5%	4%	-15%	-11%
Tc112	F	1675	1675	1675	1675	1675	1675	0%	0%	0%	0%	0%
Tc13	F	1621.2	1625.1	1627.7	1623	1624.6	1622	0%	0%	0%	0%	0%
Tc14	F	1059.6	1105.1	1058.1	1130.8	1020.6	1053.1	-4%	-4%	2%	-8%	-5%
Tc21	F	350	454.01	434.4	452.66	350	350	-23%	-4%	0%	-23%	-23%
Tc26	F	454.14	454.01	434.4	452.66	350	350	0%	-4%	0%	-23%	-23%
Tds	F	59	59	59	59	59	59	0%	0%	0%	0%	0%
Te61	F	190	190	190	190	350	350	0%	0%	0%	84%	84%
Ts36	F	700.89	700.89	700.89	700.89	700.89	700.89	0%	0%	0%	0%	0%
Ts38	F	900	900	900	900	900	900	0%	0%	0%	0%	0%
Ts39	F	115.22	115.22	566.56	115.22	115.22	115.22	0%	392%	0%	0%	0%
Ts402	F	115.22	115.22	115.22	115.22	115.22	115.22	0%	0%	0%	0%	0%
Tsuper	F	207	207	207	207	207	207	0%	0%	0%	0%	0%
Ublr	btu/hr-ft2-F	10	10	10	10	10	10	0%	0%	0%	0%	0%
Uecon	btu/hr-ft2-F	12	12	12	12	12	12	0%	0%	0%	0%	0%
Usuper	btu/hr-ft2-F	8	8	8	8	8	8	0%	0%	0%	0%	0%
Vhx1g	ft/s	63.973	74.967	79.937	69.389	68.799	62.905	-15%	7%	-7%	-8%	-16%
Vhx2g	ft/s	52.411	65.434	53.068	73.586	42.575	49.527	-20%	-19%	12%	-35%	-24%
WRaux	kW	494.65	555.16	518.1	595.62	170.48	194.61	-11%	-7%	7%	-69%	-65%
WRcmp	kW	5310.1	5310.1	5310.1	5310.1	5310.1	5310.1	0%	0%	0%	0%	0%
WRcnv	kW	12	12	12	12	12	12	0%	0%	0%	0%	0%
WRctr	kW	45	45	45	45	45	45	0%	0%	0%	0%	0%
WRdry	kW	25	25	25	25	0	0	0%	0%	0%	-100%	-100%
WRfn1	kW	45.722	45.744	48.823	45.841	0	0	0%	7%	0%	-100%	-100%
WRfn3	kW	88.93	209.23	176.53	240.6	70.798	84.252	-57%	-16%	15%	-66%	-60%
WRfn4	kW	43.999	35.223	27.002	41.737	11.322	19.581	25%	-23%	18%	-68%	-44%
WRfn5	kW	87.104	36.096	37.88	36.125	0	0	141%	5%	0%	-100%	-100%

		System Number						Percent Change from Sys 2				
Name	Units	1	2	3-2	4	5-1	5-2	1	3-2	4	5-1	5-2
WRgn1	kW	2071.2	2078.6	2646.1	2702	2064.1	2688.8	0%	27%	30%	-1%	29%
WRgn2	kW	2991.1	2988.5	2468.7	3286.7	2619.7	2915.6	0%	-17%	10%	-12%	-2%
WRgrs	kW	5062.3	5067.1	5114.8	5988.7	4683.8	5604.5	0%	1%	18%	-8%	11%
WRgtb	kW	7569.2	7577.3	8196.3	8257.3	7561.5	8242.9	0%	8%	9%	0%	9%
WRhml	kW	112.5	112.5	112.5	112.5	0	0	0%	0%	0%	-100%	-100%
WRpmp	kW	24.396	24.373	23.362	26.813	21.359	23.779	0%	-4%	10%	-12%	-2%
WRscw	kW	10	10	10	10	10	10	0%	0%	0%	0%	0%
WRstb1	kW (1st stage)	3262.6	3259.7	1090.6	3584.9	2857.4	3180.2	0%	-67%	10%	-12%	-2%
WRstb2	kW (2nd stage)	0	0	1602.2	0	0	0	ERR	ERR	ERR	ERR	ERR
Wa01	lb h2o/lb dry a	0.007	0.007	0.007	0.007	0.007	0.007	0%	0%	0%	0%	0%
Wb51	lb h2o/lb fuel	0.5	0.5	0.5	0.5	0.5	0.5	0%	0%	0%	0%	0%
Wb52	lb h2o/lb fuel	0.12	0.12	0.12	0.12	0.5	0.5	0%	0%	0%	317%	317%
XC	lb C/lb dry fue	0.523	0.523	0.523	0.523	0.523	0.523	0%	0%	0%	0%	0%
XCCO	lb C burned CO/	0.01	0.01	0.01	0.01	0.01	0.01	0%	0%	0%	0%	0%
XCCOoi	lb C burned CO/	0.01	0.01	0.01	0.01	0.01	0.01	0%	0%	0%	0%	0%
XCoil	lb C/lb oil	0.87	0.87	0.87	0.87	0.87	0.87	0%	0%	0%	0%	0%
XH2	lb H2/lb dry fu	0.063	0.063	0.063	0.063	0.063	0.063	0%	0%	0%	0%	0%
XH2oil	lb H2/lb oil	0.127	0.127	0.127	0.127	0.127	0.127	0%	0%	0%	0%	0%
XN2	lb N2/lb dry fu	0.001	0.001	0.001	0.001	0.001	0.001	0%	0%	0%	0%	0%
XN2oil	lb N2/lb oil	0.0015	0.0015	0.0015	0.0015	0.0015	0.0015	0%	0%	0%	0%	0%
XO2	lb O2/lb dry fu	0.405	0.405	0.405	0.405	0.405	0.405	0%	0%	0%	0%	0%
XO2oil	lb O2/lb oil	0.0015	0.0015	0.0015	0.0015	0.0015	0.0015	0%	0%	0%	0%	0%
Xash	lb ash/lb dry f	0.005	0.005	0.005	0.005	0.005	0.005	0%	0%	0%	0%	0%
Xashoi	lb ash/lb oil	0	0	0	0	0	0	ERR	ERR	ERR	ERR	ERR
Xdirt	lb dirt/lb dry	0.035	0.035	0.035	0.035	0.035	0.035	0%	0%	0%	0%	0%
Xs39	lb/lb quality	0.93731	0.93725	0	0.93748	0.93707	0.93727	0%	-100%	0%	0%	0%
Xs402	lb/lb quality	0.93731	0.93725	0.92677	0.93748	0.93707	0.93727	0%	-1%	0%	0%	0%
chx1		0.254	0.254	0.254	0.254	0.254	0.254	0%	0%	0%	0%	0%
chx2		0.507	0.507	0.507	0.507	0.507	0.507	0%	0%	0%	0%	0%
cmpbld	%	4	4	4	4	4	4	0%	0%	0%	0%	0%
dPap1	in. h2o	5	0	0	0	0	0	ERR	ERR	ERR	ERR	ERR
dPap2	in. h2o	5	0	0	0	0	0	ERR	ERR	ERR	ERR	ERR
dPap3	in. h2o	0	5	5	5	5	5	-100%	0%	0%	0%	0%
dPbag	in. h2o	2.5	2.5	2.5	2.5	0	0	0%	0%	0%	-100%	-100%
dPblrg	in. h2o	8	8	8	8	8	8	0%	0%	0%	0%	0%
dPblrw	psia	10	10	10	10	10	10	0%	0%	0%	0%	0%
dPcmb	in. h2o	11	11	11	11	11	11	0%	0%	0%	0%	0%
dPcmp	in. h2o	4	4	4	4	4	4	0%	0%	0%	0%	0%
dPcyc1	in. h2o	2.5	2.5	2.5	2.5	0	0	0%	0%	0%	-100%	-100%
dPcyc3	in. h2o	5	5	5	5	5	5	0%	0%	0%	0%	0%
dPdry	in. h2o	9	9	9	9	0	0	0%	0%	0%	-100%	-100%
dPfn1	in. h2o	5.7	5.7	5.7	5.7	0	0	0%	0%	0%	-100%	-100%
dPfn3	in. h2o	7.5936	14.569	12.647	15.933	6.5039	7.3193	-48%	-13%	9%	-55%	-50%
dPfn4	in. h2o	14.5	9.5	9.5	9.5	14.3	14.3	53%	0%	0%	51%	51%
dPfn5	in. h2o	8.5	3.5	3.5	3.5	0	0	143%	0%	0%	-100%	-100%
dPgtb	in. h2o	2	2	2	2	2	2	0%	0%	0%	0%	0%
dPhx1g	in. h2o	0.113	0.15146	0.16677	0.13126	0.12628	0.10699	-25%	10%	-13%	-17%	-29%
dPhx1w	in. h2o	6.8172	6.2564	5.3635	8.6147	4.4745	6.4145	9%	-14%	38%	-28%	3%
dPhx2a	in. h2o	2.991	2.6058	3.5921	2.4479	3.3656	3.0726	15%	38%	-6%	29%	18%
dPhx2g	in. h2o	3.9806	5.9175	3.9807	7.3015	2.6776	3.5123	-33%	-33%	23%	-55%	-41%

Name	Units	System Number 1	2	3-2	4	5-1	5-2	Percent Change from Sys 2: 1	3-2	4	5-1	5-2
dPtrim	psi	0	0	0	1	0	1	ERR	ERR	ERR	ERR	ERR
drylos	fract ht loss	0.05	0.05	0.05	0.05	0.05	0.05	0%	0%	0%	0%	0%
fouhx1	fouling factor	0.005	0.005	0.005	0.005	0.005	0.005	0%	0%	0%	0%	0%
fouhx2	fouling factor	0.005	0.005	0.005	0.005	0.005	0.005	0%	0%	0%	0%	0%
fshx1	factor safety	0.9	0.9	0.9	0.9	0.9	0.9	0%	0%	0%	0%	0%
fshx2	factor safety	0.9	0.9	0.9	0.9	0.9	0.9	0%	0%	0%	0%	0%
lendry	feet	37.799	37.799	37.799	37.799	0	0	0%	0%	0%	-100%	-100%
npnhx1	pipes normal	60	60	60	60	60	60	0%	0%	0%	0%	0%
npnhx2	pipes normal	72	72	72	72	72	72	0%	0%	0%	0%	0%
npphx1	pipes parallel	1	1	1	1	1	1	0%	0%	0%	0%	0%
npphx2	pipes parallel	8	8	8	8	8	8	0%	0%	0%	0%	0%
nuhx1	no. of U's	1	1	1	1	1	1	0%	0%	0%	0%	0%
nuhx2	no. of U's	3	3	3	3	3	3	0%	0%	0%	0%	0%
radlos	frac. energy in	0.02	0.02	0.02	0.02	0.02	0.02	0%	0%	0%	0%	0%
radtrm	frac. HHVoil	0.02	0.02	0.02	0.02	0.02	0.02	0%	0%	0%	0%	0%
rufhx1		0.002	0.002	0.002	0.002	0.002	0.002	0%	0%	0%	0%	0%
rufhx2		0.002	0.002	0.002	0.002	0.002	0.002	0%	0%	0%	0%	0%
typhx1	1 = in-line	1	1	1	1	1	1	0%	0%	0%	0%	0%
typhx2	2 = staggered	2	2	2	2	2	2	0%	0%	0%	0%	0%
unbcbn	lb/lb	0.02	0.02	0.02	0.02	0.02	0.02	0%	0%	0%	0%	0%
xhx1		0.632	0.632	0.632	0.632	0.632	0.632	0%	0%	0%	0%	0%
xhx2		0.564	0.564	0.564	0.564	0.564	0.564	0%	0%	0%	0%	0%
ya01Ar	mol fract Ar	0.00919	0.00919	0.00919	0.00919	0.00919	0.00919	0%	0%	0%	0%	0%
ya01CD	mol fract CO2	0.00029	0.00029	0.00029	0.00029	0.00029	0.00029	0%	0%	0%	0%	0%
ya01CO	mol fract CO	0	0	0	0	0	0	ERR	ERR	ERR	ERR	ERR
ya01N2	mol fract N2	0.77221	0.77221	0.77221	0.77221	0.77221	0.77221	0%	0%	0%	0%	0%
ya01O2	mol fract O2	0.20717	0.20717	0.20717	0.20717	0.20717	0.20717	0%	0%	0%	0%	0%
ya01W	mol fract H2O	0.01112	0.01112	0.01112	0.01112	0.01112	0.01112	0%	0%	0%	0%	0%
ya02Ar	mol fract Ar	0.00919	0.00919	0.00851	0.00919	0.00919	0.00919	0%	-7%	0%	0%	0%
ya02CD	mol fract CO2	0.00029	0.00029	0.00027	0.00029	0.00029	0.00029	0%	-7%	0%	0%	0%
ya02CO	mol fract CO	0	0	0	0	0	0	ERR	ERR	ERR	ERR	ERR
ya02N2	mol fract N2	0.77221	0.77221	0.71497	0.77221	0.77221	0.77221	0%	-7%	0%	0%	0%
ya02O2	mol fract O2	0.20717	0.20717	0.19181	0.20717	0.20717	0.20717	0%	-7%	0%	0%	0%
ya02W	mol fract H2O	0.01112	0.01112	0.08442	0.01112	0.01112	0.01112	0%	659%	0%	0%	0%
ya03Ar	mol fract Ar	0.00919	0.00919	0.00851	0.00917	0.00919	0.00917	0%	-7%	0%	0%	0%
ya03CD	mol fract CO2	0.00029	0.00029	0.00027	0.00594	0.00029	0.00594	0%	-7%	1903%	0%	1903%
ya03CO	mol fract CO	0	0	0	0.00005	0	0.00005	ERR	ERR	ERR	ERR	ERR
ya03N2	mol fract N2	0.77221	0.77221	0.71497	0.77027	0.77221	0.77027	0%	-7%	0%	0%	0%
ya03O2	mol fract O2	0.20717	0.20717	0.19181	0.1985	0.20717	0.1985	0%	-7%	-4%	0%	-4%
ya03W	mol fract H2O	0.01112	0.01112	0.08442	0.01606	0.01112	0.01606	0%	659%	44%	0%	44%
ya04Ar	mol fract Ar	0.00919	0.00919	0.00854	0.00917	0.00919	0.00917	0%	-7%	0%	0%	0%
ya04CD	mol fract CO2	0.00029	0.00029	0.00027	0.00571	0.00029	0.00571	0%	-7%	1827%	0%	1827%
ya04CO	mol fract CO	0	0	0	0.00005	0	0.00005	ERR	ERR	ERR	ERR	ERR
ya04N2	mol fract N2	0.77221	0.77221	0.7171	0.77035	0.77221	0.77035	0%	-7%	0%	0%	0%
ya04O2	mol fract O2	0.20717	0.20717	0.19238	0.19884	0.20717	0.19884	0%	-7%	-4%	0%	-4%
ya04W	mol fract H2O	0.01112	0.01112	0.08170	0.01586	0.01112	0.01586	0%	634%	43%	0%	43%
yc11Ar	mol fract Ar	0.00866	0.00866	0.00808	0.00864	0.00805	0.00803	0%	-7%	0%	-7%	-7%
yc11CD	mol fract CO2	0.06736	0.06736	0.06281	0.07229	0.06259	0.06719	0%	-7%	7%	-7%	0%
yc11CO	mol fract CO	0.00069	0.00069	0.00064	0.00074	0.00064	0.00068	0%	-7%	7%	-7%	0%
yc11N2	mol fract N2	0.72798	0.72798	0.67881	0.72635	0.67648	0.67508	0%	-7%	0%	-7%	-7%

Name	Units	System Number 1	2	3-2	4	5-1	5-2	Percent Change from Sys 2: 1	3-2	4	5-1	5-2
yc1102	mol fract O2	0.12314	0.12314	0.11482	0.11552	0.11443	0.10736	0%	-7%	-6%	-7%	-13%
yc11W	mol fract H2O	0.07214	0.07214	0.13482	0.07644	0.1378	0.14163	0%	87%	6%	91%	96%
ydsAr		0.0092	0.0092	0.0092	0.0092	0.0092	0.0092	0%	0%	0%	0%	0%
ydsCD		0.0003	0.0003	0.0003	0.0003	0.0003	0.0003	0%	0%	0%	0%	0%
ydsCO		0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0%	0%	0%	0%	0%
ydsN2		0.7722	0.7722	0.7722	0.7722	0.7722	0.7722	0%	0%	0%	0%	0%
ydsO2		0.2072	0.2072	0.2072	0.2072	0.2072	0.2072	0%	0%	0%	0%	0%
ydsW		0.0111	0.0111	0.0111	0.0111	0.0111	0.0111	0%	0%	0%	0%	0%
ye61Ar	mol fract Ar	0.00805	0.00805	0.00754	0.00803	0.00805	0.00803	0%	-6%	0%	0%	0%
ye61CD	mol fract CO2	0.06259	0.06259	0.05865	0.06719	0.06259	0.06719	0%	-6%	7%	0%	7%
ye61CO	mol fract CO	0.00064	0.00064	0.00060	0.00068	0.00064	0.00068	0%	-6%	7%	0%	7%
ye61N2	mol fract N2	0.67645	0.67648	0.63381	0.67508	0.67648	0.67508	0%	-6%	0%	0%	0%
ye61O2	mol fract O2	0.11442	0.11443	0.10721	0.10736	0.11443	0.10736	0%	-6%	-6%	0%	-6%
ye61W	mol fract H2O	0.13783	0.1378	0.19218	0.14163	0.1378	0.14163	0%	39%	3%	0%	3%

		afrat	afrat	afrat	Ta032	Ta032	Ta032	Ta032
AFrat	lb air/lb fuel	12	17	22	17	17	17	17
Abl.	square feet	19331	15700	15330	15218	15700	15999	16968
Ahx1	square feet	459.59	220.51	130.04	190.2	220.51	242.44	295.57
Ahx2	square feet	6827.8	8160.9	10653	8929	8160.9	7816.3	7017.9
CFMfn1	cu ft/min	56424	53658	52149	53544	53658	53720	53895
CFMfn3	cu ft/min	79646	61678	52647	58327	61678	63509	68889
CFMfn4	cu ft/min	22121	7337.1	0	4242.2	7337.1	9021.8	13948
CFMfn5	cu ft/min	56424	53658	52149	53544	53658	53720	53895
EFFHg	% gross, HHV	22.881	24.586	25.834	22.626	24.586	25.558	27.972
EFFHn	% net, HHV	22.016	23.732	24.955	21.802	23.732	24.686	27.033
EFFL1g	% gross, LHV1	24.533	26.357	27.69	24.272	26.357	27.39	29.954
EFFL3g	% gross, LHV3	27.899	29.86	31.26	27.877	29.86	30.833	33.219
EFFcmp	%	83.2	83.2	83.2	83.2	83.2	83.2	83.2
EFFfn1	%	63	63	63	63	63	63	63
EFFfn3	%	63	63	63	63	63	63	63
EFFfn4	%	63	63	63	63	63	63	63
EFFfn5	%	63	63	63	63	63	63	63
EFFgb1	%	96	96	96	96	96	96	96
EFFgb2	%	96	96	96	96	96	96	96
EFFgn1	%	95.5	95.5	95.5	95.5	95.5	95.5	95.5
EFFgn2	%	95.5	95.5	95.5	95.5	95.5	95.5	95.5
EFFgtb	%	91.2	91.2	91.2	91.2	91.2	91.2	91.2
EFFpmp	%	68	68	68	68	68	68	68
EFFstb	%	80	80	80	80	80	80	80
HHV	btu/lb	8800	8800	8800	8800	8800	8800	8800
HHVoil	btu/lb	19870	19870	19870	19870	19870	19870	19870
IDhx1	inches	0.56	0.56	0.56	0.56	0.56	0.56	0.56
IDhx2	inches	0.81	0.81	0.81	0.81	0.81	0.81	0.81
IRRblr	kW	2360.3	1301.9	744.48	1119.6	1301.9	1402.9	1703.6
IRRcmb	kW	12840	9491.9	7578.3	9724	9491.9	9384.6	9127.9
IRRcmp	kW	665.62	665.62	665.62	665.62	665.62	665.62	665.62
IRRcnd	kW	1192.8	751.06	517	674.49	751.06	792.96	915.9
IRRdry	kW	0	0	0	0	0	0	0
IRRfn1	kW	0	0	0	0	0	0	0
IRRfn3	kW	40.109	31.173	27.99	26.195	31.173	34.265	45.11
IRRfn4	kW	21.843	7.2451	0	4.189	7.2451	8.9087	13.773
IRRfn5	kW	0	0	0	0	0	0	0
IRRgn1	kW	244.98	244.01	242.02	187.32	244.01	275.97	367.51
IRRgn2	kW	407.39	264.6	182.15	237.73	264.6	279.26	322.16
IRRgtb	kW	463.81	463.06	461.52	461.8	463.06	465.08	472.2
IRRhx1	kW	371.19	145.32	99.094	129.47	145.32	154.39	182.35
IRRhx2	kW	730.72	644.49	542.59	619.12	644.49	657.69	694.47
IRRinj	kW	0	0	0	0	0	0	0
IRRmxc	kW (mix ex cmb)	1144.7	337.39	0	189.34	337.39	421.31	679.18
IRRmxd	kW (mix in dry)	0	0	0	0	0	0	0
IRRpmp	kW	11.93	7.6093	5.238	6.835	7.6093	8.0326	9.2731
IRRst1	kW (1st stage)	1104.4	717.31	493.8	644.49	717.31	757.05	873.37
IRRst2	kW (2nd stage)	0	0	0	0	0	0	0
IRRtrm	kW	731.25	731.26	731.27	0	731.26	1084.4	2022.2
Kphx1	btu/hr-ft-F	20	20	20	20	20	20	20
Kphx2	btu/hr-ft-F	20	20	20	20	20	20	20
LHV1	BTU/lb -h2ocomb	8203.2	8203.2	8203.2	8203.2	8203.2	8203.2	8203.2

		afrat	afrat	afrat	Ta032	Ta032	Ta032	Ta032
LHV2	BTU/lb -h2owood	7142.2	7142.2	7142.2	7142.2	7142.2	7142.2	7142.2
LHVoil	btu/lb	18660	18660	18660	18660	18660	18660	18660
MRa021	lb/s	36.679	36.679	36.679	36.679	36.679	36.679	36.679
MRa022	lb/s	36.679	36.679	36.679	36.679	36.679	36.679	36.679
MRa032	lb/s	36.78	36.78	36.78	36.679	36.78	36.834	36.989
MRa04	lb/s	38.308	38.308	38.308	38.208	38.308	38.363	38.517
MRa06	lb/s	1.5283	1.5283	1.5283	1.5283	1.5283	1.5283	1.5283
MRb51	lb/s	6.432	4.5402	3.5084	4.5419	4.5402	4.5394	4.5368
MRb52	lb/s	6.432	4.5402	3.5084	4.5419	4.5402	4.5394	4.5368
MRbdwd	lb/s	3.1607	2.2311	1.724	2.2319	2.2311	2.2306	2.2294
MRc111	lb/s	44.573	42.731	41.726	42.63	42.731	42.786	42.941
MRc112	lb/s	62.354	48.676	41.726	46.067	48.676	50.097	54.249
MRc12	lb/s	62.354	48.676	41.726	46.067	48.676	50.097	54.249
MRc16	lb/s	0	0	0	0	0	0	0
MRc16m	lb/s	99999	99999	99999	99999	99999	99999	99999
MRc231	lb/s	17.781	5.9454	0	3.4367	5.9454	7.3116	11.308
MRc23m	lb/s	0	1	1	1	1	1	1
MRc25	lb/s	44.573	42.731	41.726	42.63	42.731	42.786	42.941
MRc26	lb/s	44.573	42.731	41.726	42.63	42.731	42.786	42.941
MRe61	lb/s	44.573	42.731	41.726	42.63	42.731	42.786	42.941
MRoil	lb/s	0.10036	0.10036	0.10036	0	0.10036	0.15512	0.30991
MRs31	lb/s	11.824	7.5418	5.1916	6.7743	7.5418	7.9613	9.1908
MRs402	lb/s	11.824	7.5418	5.1917	6.7743	7.5418	7.9613	9.1909
MRstp	lb/s	38.75	38.75	38.75	38.75	38.75	38.75	38.75
Ma01	lb/mol	28.845	28.845	28.845	28.845	28.845	28.845	28.845
Ma02	lb/mol	28.845	28.845	28.845	28.845	28.845	28.845	28.845
Ma03	lb/mol	28.851	28.851	28.851	28.845	28.851	28.854	28.863
ODhx1	inches	0.75	0.75	0.75	0.75	0.75	0.75	0.75
ODhx2	inches	1	1	1	1	1	1	1
Pa01	psia	14.696	14.696	14.696	14.696	14.696	14.696	14.696
Pa021	psia	132.42	132.42	132.42	132.42	132.42	132.42	132.42
Pa022	psia	132.42	132.42	132.42	132.42	132.42	132.42	132.42
Pa031	psia	129.85	129.35	128.31	129.05	129.35	129.48	129.78
Pa032	psia	128.85	128.35	127.31	129.05	128.35	128.48	128.78
Pa04	in. h2o	26	26	26	26	26	26	26
Pc111	in. h2o	15	15	15	15	15	15	15
Pc112	in. h2o	15	15	15	15	15	15	15
Pc113	in. h2o	10	10	10	10	10	10	10
Pc13	in. h2o	9.9563	9.893	9.7961	9.8737	9.893	9.905	9.9233
Pc14	in. h2o	6.4072	6.3807	6.0007	7.1961	6.3807	5.8868	4.2172
Pc211	in. h2o	-1.5928	-1.6193	-1.9993	-0.80386	-1.6193	-2.1132	-3.7828
Pc212	in. h2o	-6.5928	-6.6193	-6.9993	-5.8039	-6.6193	-7.1132	-8.7828
Pc22	in. h2o	0.7	0.7	0.7	0.7	0.7	0.7	0.7
Pc232	in. h2o	15	15	15	15	15	15	15
Pc26	in. h2o	0.7	0.7	0.7	0.7	0.7	0.7	0.7
Pc27	in. h2o	0.7	0.7	0.7	0.7	0.7	0.7	0.7
Pc29	in. h2o	0.7	0.7	0.7	0.7	0.7	0.7	0.7
Pds	psia	14.696	14.696	14.696	14.696	14.696	14.696	14.696
Pe61	in. h2o	0.7	0.7	0.7	0.7	0.7	0.7	0.7
Pe62	in. h2o	0.7	0.7	0.7	0.7	0.7	0.7	0.7
Pe63	in. h2o	0.7	0.7	0.7	0.7	0.7	0.7	0.7
Pe64	in. h2o	0.7	0.7	0.7	0.7	0.7	0.7	0.7

		afrat	afrat	afrat	Ta032	Ta032	Ta032	Ta032
Prcinj	% injection	0	0	0	0	0	0	0
Prcmp	press. ratio	9.1	9.1	9.1	9.1	9.1	9.1	9.1
Ps31	psia	650	650	650	650	650	650	650
Ps36	psia	640	640	640	640	640	640	640
Ps38	psia	520.21	633.59	633.96	635.53	633.59	632.15	627.24
Ps39	psia	1.5	1.5	1.5	1.5	1.5	1.5	1.5
Ps402	psia	1.5	1.5	1.5	1.5	1.5	1.5	1.5
Qcmb	btu/s	27814	19633	15171	19641	19633	19630	19619
Qcnd	btu/s	11559	7278	5009.9	6536	7278	7684	8875.3
SnDhx1		2	2	2	2	2	2	2
SnDhx2		1.7	1.7	1.7	1.7	1.7	1.7	1.7
SpDhx1		2	2	2	2	2	2	2
SpDhx2		1.5	1.5	1.5	1.5	1.5	1.5	1.5
Ta01	F	59	59	59	59	59	59	59
Ta021	F	595.52	595.52	595.52	595.52	595.52	595.52	595.52
Ta022	F	595.52	595.52	595.52	595.52	595.52	595.52	595.52
Ta031	F	1450	1450	1450	1450	1450	1450	1450
Ta032	F	1612	1612	1612	1450	1612	1700	1950
Ta04	F	822.31	823.41	825.69	715.39	823.41	881.46	1047.9
Tc111	F	2155.3	1845.2	1652.6	1773.9	1845.2	1883.6	1994.9
Tc112	F	1675	1675	1652.6	1675	1675	1675	1675
Tc13	F	1608.9	1622	1609.1	1624.6	1622	1620.7	1617.3
Tc14	F	1180.5	1053.1	928.57	1020.6	1053.1	1069.3	1111.5
Tc21	F	350	350	350	350	350	350	350
Tc26	F	350	350	350	350	350	350	350
Tds	F	59	59	59	59	59	59	59
Te61	F	350	350	350	350	350	350	350
Ts36	F	700.89	700.89	700.89	700.89	700.89	700.89	700.89
Ts38	F	900	900	900	900	900	900	900
Ts39	F	115.22	115.22	115.22	115.22	115.22	115.22	115.22
Ts402	F	115.22	115.22	115.22	115.22	115.22	115.22	115.22
Tsuper	F	207	207	207	207	207	207	207
Ublr	btu/hr-ft2-F	10	10	10	10	10	10	10
Uecon	btu/hr-ft2-F	12	12	12	12	12	12	12
Usuper	btu/hr-ft2-F	8	8	8	8	8	8	8
Vhx1g	ft/s	38.916	62.905	88.546	68.799	62.905	58.996	52.559
Vhx2g	ft/s	59.258	49.527	42.632	42.575	49.527	53.387	64.923
WRaux	kW	271.72	194.61	159.02	170.48	194.61	208.79	255.12
WRcmp	kW	5310.1	5310.1	5310.1	5310.1	5310.1	5310.1	5310.1
WRcnv	kW	12	12	12	12	12	12	12
WRctr	kW	45	45	45	45	45	45	45
WRdry	kW	0	0	0	0	0	0	0
WRfn1	kW	0	0	0	0	0	0	0
WRfn3	kW	108.4	84.252	75.649	70.798	84.252	92.608	121.92
WRfn4	kW	59.036	19.581	0	11.322	19.581	24.078	37.223
WRfn5	kW	0	0	0	0	0	0	0
WRgn1	kW	2699.5	2688.8	2666.8	2064.1	2688.8	3041	4049.7
WRgn2	kW	4489.1	2915.6	2007.2	2619.7	2915.6	3077.2	3550
WRgrs	kW	7188.6	5604.5	4674	4683.8	5604.5	6118.2	7599.7
WRgtb	kW	8254.6	8242.9	8219	7561.5	8242.9	8627	9727.3
WRhml	kW	0	0	0	0	0	0	0
WRpmp	kW	37.281	23.779	16.369	21.359	23.779	25.102	28.979

	afrat	afrat	afrat	Ta032	Ta032	Ta032	Ta032
WRscw kW	10	10	10	10	10	10	10
WRstb1 kW (1st stage)	4896.5	3180.2	2189.3	2857.4	3180.2	3356.5	3872.2
WRstb2 kW (2nd stage)	0	0	0	0	0	0	0
Wa01 lb h2o/lb dry a	0.007	0.007	0.007	0.007	0.007	0.007	0.007
Wb51 lb h2o/lb fuel	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Wb52 lb h2o/lb fuel	0.5	0.5	0.5	0.5	0.5	0.5	0.5
XC lb C/lb dry fue	0.523	0.523	0.523	0.523	0.523	0.523	0.523
XCCO lb C burned CO/	0.01	0.01	0.01	0.01	0.01	0.01	0.01
XCCOoi lb C burned CO/	0.01	0.01	0.01	0.01	0.01	0.01	0.01
XCoil lb C/lb oil	0.87	0.87	0.87	0.87	0.87	0.87	0.87
XH2 lb H2/lb dry fu	0.063	0.063	0.063	0.063	0.063	0.063	0.063
XH2oil lb H2/lb oil	0.127	0.127	0.127	0.127	0.127	0.127	0.127
XN2 lb N2/lb dry fu	0.001	0.001	0.001	0.001	0.001	0.001	0.001
XN2oil lb N2/lb oil	0.0015	0.0015	0.0015	0.0015	0.0015	0.0015	0.0015
XO2 lb O2/lb dry fu	0.405	0.405	0.405	0.405	0.405	0.405	0.405
XO2oil lb O2/lb oil	0.0015	0.0015	0.0015	0.0015	0.0015	0.0015	0.0015
Xash lb ash/lb dry f	0.005	0.005	0.005	0.005	0.005	0.005	0.005
Xashoi lb ash/lb oil	0	0	0	0	0	0	0
Xdirt lb dirt/lb dry	0.035	0.035	0.035	0.035	0.035	0.035	0.035
Xs39 lb/lb quality	0.94946	0.93727	0.93723	0.93707	0.93727	0.93741	0.93789
Xs402 lb/lb quality	0.94946	0.93727	0.93723	0.93707	0.93727	0.93741	0.93789
chx1	0.254	0.254	0.254	0.254	0.254	0.254	0.254
chx2	0.507	0.507	0.507	0.507	0.507	0.507	0.507
cmpbld %	4	4	4	4	4	4	4
dPap1 in. h2o	0	0	0	0	0	0	0
dPap2 in. h2o	0	0	0	0	0	0	0
dPap3 in. h2o	5	5	5	5	5	5	5
dPbag in. h2o	0	0	0	0	0	0	0
dPblrg in. h2o	8	8	8	8	8	8	8
dPblrw psia	10	10	10	10	10	10	10
dPcmb in. h2o	11	11	11	11	11	11	11
dPcmp in. h2o	4	4	4	4	4	4	4
dPcyc1 in. h2o	0	0	0	0	0	0	0
dPcyc3 in. h2o	5	5	5	5	5	5	5
dPdry in. h2o	0	0	0	0	0	0	0
dPfn1 in. h2o	0	0	0	0	0	0	0
dPfn3 in. h2o	7.2928	7.3193	7.6993	6.5039	7.3193	7.8132	9.4828
dPfn4 in. h2o	14.3	14.3	14.3	14.3	14.3	14.3	14.3
dPfn5 in. h2o	0	0	0	0	0	0	0
dPgtb in. h2o	2	2	2	2	2	2	2
dPhx1g in. h2o	0.043672	0.10699	0.20389	0.12628	0.10699	0.095014	0.076737
dPhx1w in. h2o	119.79	6.4145	6.0414	4.4745	6.4145	7.8537	12.755
dPhx2a in. h2o	2.5657	3.0726	4.1091	3.3656	3.0726	2.9414	2.6379
dPhx2g in. h2o	3.5491	3.5123	3.7954	2.6776	3.5123	4.0182	5.7061
dPtrim psi	1	1	1	0	1	1	1
drylos fract ht loss	0.05	0.05	0.05	0.05	0.05	0.05	0.05
fouhx1 fouling factor	0.005	0.005	0.005	0.005	0.005	0.005	0.005
fouhx2 fouling factor	0.005	0.005	0.005	0.005	0.005	0.005	0.005
fshx1 factor safety	0.9	0.9	0.9	0.9	0.9	0.9	0.9
fshx2 factor safety	0.9	0.9	0.9	0.9	0.9	0.9	0.9
lendry feet	0	0	0	0	0	0	0
npnhx1 pipes normal	40	60	40	60	60	60	60

	afrat	afrat	afrat	Ta032	Ta032	Ta032	Ta032
npnhx2 pipes normal	96	72	52	72	72	72	72
npphx1 pipes parallel	1	1	1	1	1	1	1
npphx2 pipes parallel	6	8	11	8	8	8	8
nuhx1 no. of U's	1	1	1	1	1	1	1
nuhx2 no. of U's	3	3	3	3	3	3	3
radlos frac. energy in	0.02	0.02	0.02	0.02	0.02	0.02	0.02
radtrm frac. HHVoil	0.02	0.02	0.02	0.02	0.02	0.02	0.02
rufhx1	0.002	0.002	0.002	0.002	0.002	0.002	0.002
rufhx2	0.002	0.002	0.002	0.002	0.002	0.002	0.002
typhx1 1 = in-line	1	1	1	1	1	1	1
typhx2 2 = staggered	2	2	2	2	2	2	2
unbcbn lb/lb	0.02	0.02	0.02	0.02	0.02	0.02	0.02
xhx1	0.632	0.632	0.632	0.632	0.632	0.632	0.632
xhx2	0.564	0.564	0.564	0.564	0.564	0.564	0.564
ya01Ar mol fract Ar	0.009196	0.009196	0.009196	0.009196	0.009196	0.009196	0.009196
ya01CD mol fract CO2	0.000296	0.000296	0.000296	0.000296	0.000296	0.000296	0.000296
ya01CO mol fract CO	0	0	0	0	0	0	0
ya01N2 mol fract N2	0.77221	0.77221	0.77221	0.77221	0.77221	0.77221	0.77221
ya01O2 mol fract O2	0.20717	0.20717	0.20717	0.20717	0.20717	0.20717	0.20717
ya01W mol fract H2O	0.011129	0.011129	0.011129	0.011129	0.011129	0.011129	0.011129
ya02Ar mol fract Ar	0.009196	0.009196	0.009196	0.009196	0.009196	0.009196	0.009196
ya02CD mol fract CO2	0.000296	0.000296	0.000296	0.000296	0.000296	0.000296	0.000296
ya02CO mol fract CO	0	0	0	0	0	0	0
ya02N2 mol fract N2	0.77221	0.77221	0.77221	0.77221	0.77221	0.77221	0.77221
ya02O2 mol fract O2	0.20717	0.20717	0.20717	0.20717	0.20717	0.20717	0.20717
ya02W mol fract H2O	0.011129	0.011129	0.011129	0.011129	0.011129	0.011129	0.011129
ya03Ar mol fract Ar	0.009173	0.009173	0.009173	0.009196	0.009173	0.009160	0.009125
ya03CD mol fract CO2	0.005941	0.005941	0.005941	0.000296	0.005941	0.009008	0.017636
ya03CO mol fract CO	0.000057	0.000057	0.000057	0	0.000057	0.000088	0.000175
ya03N2 mol fract N2	0.77027	0.77027	0.77027	0.77221	0.77027	0.76922	0.76625
ya03O2 mol fract O2	0.1985	0.1985	0.1985	0.20717	0.1985	0.19378	0.18053
ya03W mol fract H2O	0.016061	0.016061	0.016061	0.011129	0.016061	0.018741	0.026278
ya04Ar mol fract Ar	0.009174	0.009174	0.009174	0.009196	0.009174	0.009162	0.009128
ya04CD mol fract CO2	0.005715	0.005715	0.005715	0.000296	0.005715	0.008661	0.016947
ya04CO mol fract CO	0.000054	0.000054	0.000054	0	0.000054	0.000084	0.000168
ya04N2 mol fract N2	0.77035	0.77035	0.77035	0.77221	0.77035	0.76934	0.76649
ya04O2 mol fract O2	0.19884	0.19884	0.19884	0.20717	0.19884	0.19432	0.18159
ya04W mol fract H2O	0.015864	0.015864	0.015864	0.011129	0.015864	0.018437	0.025676
yc11Ar mol fract Ar	0.007645	0.008039	0.008271	0.008055	0.008039	0.008030	0.008004
yc11CD mol fract CO2	0.08854	0.067194	0.054596	0.062599	0.067194	0.069693	0.076726
yc11CO mol fract CO	0.000909	0.000689	0.000559	0.000642	0.000689	0.000714	0.000785
yc11N2 mol fract N2	0.64201	0.67508	0.6946	0.67648	0.67508	0.67433	0.67219
yc11O2 mol fract O2	0.075601	0.10736	0.12611	0.11443	0.10736	0.10352	0.092705
yc11W mol fract H2O	0.1853	0.14163	0.11586	0.1378	0.14163	0.14372	0.14959
ydsAr	0.0092	0.0092	0.0092	0.0092	0.0092	0.0092	0.0092
ydsCD	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003
ydsCO	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001
ydsN2	0.7722	0.7722	0.7722	0.7722	0.7722	0.7722	0.7722
ydsO2	0.2072	0.2072	0.2072	0.2072	0.2072	0.2072	0.2072
ydsW	0.0111	0.0111	0.0111	0.0111	0.0111	0.0111	0.0111
ye61Ar mol fract Ar	0.007645	0.008039	0.008271	0.008055	0.008039	0.008030	0.008004
ye61CD mol fract CO2	0.08854	0.067194	0.054596	0.062599	0.067194	0.069693	0.076726

	afrat	afrat	afrat	Ta032	Ta032	Ta032	Ta032
ye61CO mol fract CO	0.000909	0.000689	0.000559	0.000642	0.000689	0.000714	0.000785
ye61N2 mol fract N2	0.64201	0.67508	0.6946	0.67648	0.67508	0.67433	0.67219
ye61O2 mol fract O2	0.075601	0.10736	0.12611	0.11443	0.10736	0.10352	0.092705
ye61W mol fract H2O	0.1853	0.14163	0.11586	0.1378	0.14163	0.14372	0.14959