AN ABSTRACT OF THE THESIS OF

Eric Richard Schild for the degree of Master of Science in Industrial Engineering presented on July 23, 1984

Title: A Feasibility Study of a Computer-Assisted Plywood Production Scheduling System

Abstract approved: Redacted for privacy

Dr. Edward McDowell

Plywood production scheduling for years has been considered, more or less, a sacred activity in which only a select group of individuals with the experience and knowledge could take part. With the advancements in computer technology and related fields, the plywood industry is in a position where more people can become involved in an otherwise restricted area of production planning.

The objective of this study was to design a computer-assisted scheduling package that feasibly schedules jobs through the four operations of a dry-end department.

The program, which satisfies this objective, was modeled after a local plywood plant. Though, because of

the size of the facility and the limitations of the mini-computer on which this program was developed, only a fraction of the plants entire product line was used.

The analysis of the program concentrates on evaluating the feasibility issue. That is, two trial runs were conducted to determine whether the program could generate feasible schedules. The test results for both runs indicated that the computed schedules were feasible. However, this does not warrant its practical use in an actual mill environment. Further analysis indicated that the current heuristic for scheduling the patcher, jointer, and dryer operations is not suitable for ensuring satisfactory levels of machine utilization. One possible resolution that was cited is to reformulate this heuristic so that makespan is minimized.

A Feasibility Study of a Computer-Assisted Plywood Production Scheduling System

by

Eric R. Schild

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Completed July 23, 1984

Commencement June, 1985

APPROVED:

Redacted for privacy

Associate Professor of Industrial Engineering in charge of major

Redacted for privacy

Head of Department of Industrial and General Engineering

Redacted for privacy

Dean of Graduate School

Date thesis is presented July 23, 1984

Typed by the author

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

A FEASIBILITY STUDY OF A COMPUTER-ASSISTED PLYWOOD PRODUCTION SCHEDULING SYSTEM

CHAPTER I

INTRODUCTION

1.1 BACKGROUND

Production scheduling in plywood manufacturing is primarily concerned with ordering tasks through a three phase production sequence (green-end, dry-end, and finish-end) to prevent task lateness and ensure continuous job throughput. Note, however, that both these measures are interrelated. That is, without a continuous throughput, job lateness may be inevitable. Consequently, management must pay special attention to those factors affecting product flow when developing a viable job schedule. Because of the complexity of the system, total enumeration of these factors can be quite difficult when developing a task schedule. This is especially true for mills that still schedule only from past experience.

Yaptenco and Wylie [1970] wrote that the heart of the scheduling problem appears to be limited to only two phases, the dry-end and green-end. Their justification is based on the fact that these two departments are the most critical phases of plywood manufacturing since "choices in sequences have to be made at each production center from among a number of alternatives." These choices will be based on the composition of the order file and the veneer and product inventories. They continued by stating that the source of the problem is one of timing veneer preparation so that the right type and quantities are available at the time they are needed at a specific operation. However in terms of a mill environment such an undertaking may not be easily accomplished because of the complexity imposed by the system. This complexity stems from the fact that plywood is constructed through successive operations and in most cases these operations consist of m parallel and identical processors. Therefore, another scheduling problem arises when determining whether a given operation will be utilized to its full potential or will be under utilized. Specifically, if we do not elect to use all m identical processors in an operation, what affect will this have on the in-process inventories and the throughput of succeeding operations that use these inventories? On the other hand, if we choose to run all m processors, will the in-process inventory level ever exceed the maximum storage capacity of the system?

In terms of management policy many of the answers to these questions are based on some intuitive cost benefit parameter. That is, if we choose to run all m machines, the benefits associated with improving product flow and minimizing job lateness must override the cost of employing the extra labor and machines. Unfortunately, as can be seen, this measure is not easily quantified. In most situations, such an activity is left to management personnel whose experience allows them to resolve these difficult questions about production scheduling. This approach to scheduling, as many managers will contend, has historically proven to be the best and most economical method. But since the inception of the digital computer, this means can no longer be considered best or most economical. Currently, the computer software market is being deluged with packages that assist managers in fields such as forecasting, production control, and accounting to name a few. These tools are not marketed as cure-alls but as practical real world methods to aid managers in their daily decision activities. Unfortunately, no software package to date has been written solely for the purpose of aiding plywood managers in scheduling their facilities.

1.2 OBJECTIVE

To resolve this dilemma, the objective of this study is to develop a computer package that will schedule a set of plywood jobs in a feasible manner through the dry-end phase of a plywood operation. The main emphasis of this project is to construct a program that will schedule four operations of a dry-end department so that a mill manager can see the affects that his policies as well as manufacturing restrictions have on product flow, machine utilization, and inventory when scheduling a given set of orders. The approach to be used in developing this decision-aid will attempt to integrate current management policy with known scheduling and inventory control techniques. It will be written in FORTRAN-66 and because of the increasing useage rate of mini-computers in the work place, this package will be run on a Digital PDP 11/23.

The anticipated benefits of building this package includes better understanding of system performance and as a direct result improved scheduling.

1.3 THE PLANT

1.3.1 Dry-end Description

In order to build this package, the system will be modeled using a local plywood mill (Champion Building Products, Lebanon, OR). This plant (see figure 1.1) is a relatively large facility employing nearly 450 people and producing on the average nearly 3 1/2 million surface feet of plywood per week. Presently, it is working under a curtailed work schedule and for economic considerations is only producing a limited selection of their total product

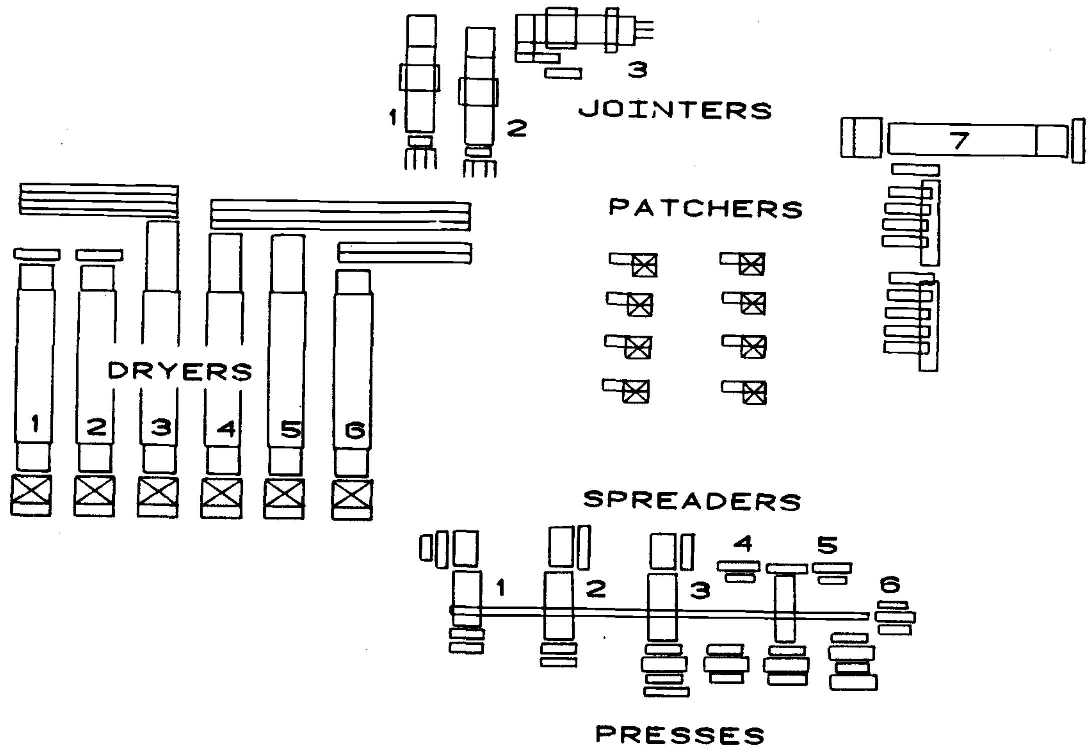


Figure 1.1 Dry-end facility lay-out

line. The dry-end department is considered typical of most mills this size. It will be modeled as a four stage job shop (see figure 1.2) where the green veneer is dried, upgraded, and laid up into plywood panels. In the first

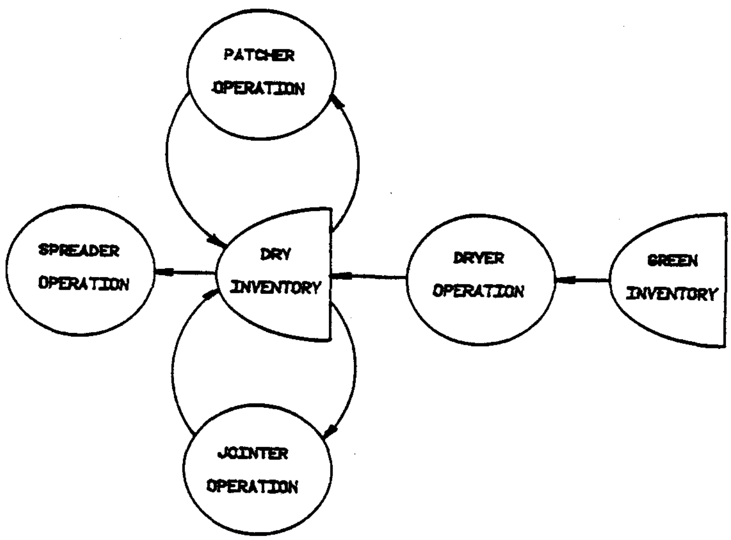


Figure 1.2 Process flow diagram

stage, the green veneer is dried on one of seven cross-flow type dryers. It is subsequently given an appearance grade

TABLE 1.1 Dry veneer grade development

WIDTH:	Random/Ft	27	54
	BP	BP	BP
	C	C	C
	D	D	D
	X	X	X
		CP	
			ECLR
			DCLR
	SOL(Ft)		A(solid)
			B

Ft-fishtails (4′ length)

and placed in dry inventory (see table 1.1). In the second and third stages, selected dry veneers are upgraded either by splicing two or more narrow sheets together with string or glue, or the appearance grades are improved by patching the noticeable defects. These two stages are commonly referred to as the jointer and patcher operations respectively. In each operation, as is true for the dryer operation, several machines are available for processing the same task. For the jointer operation, five machines are available where as the patcher operation consists of 18 machines. In the final stage, the panels are laid up using selected grades of veneer from dry inventory. This process is accomplished using six roller type spreaders. One unique aspect of this plant is its capability to manufacture variable sized panels. That is, they not only can produce a four by eight foot panel, but are also capable of manufacturing a panel as small as three by eight and as large as five by 12.

1.3.2 Management Constraints

Because of this panel size variability, the green veneer requirements increase drastically over the normal requirements for a four by eight panel. Typically, mills categorize green veneer based on thickness, width, species, and wood type. For this plant, veneer length is added to this list (see table 1.2) to allow for the variable panel

TABLE 1.2 Dry veneer categories

THICKNESS	WIDTH	LENGTH	SPECIES	WOOD TYPE
1/12"	RW	4'	Douglas-fir	sapwood
1/10"	27"	8'	hemlock*	heartwood*
1/8"*	54"	9'	noble fir*	
1/7"*		10'		
1/6"		12'		
7/32"				

*-not included in the model.

sizes. As a result of this broad categorization, processing constraints have been imposed on machines within each operation because of either machine limitations or management policy.

For the dryer operation, the limitations are included in table 1.3. Note that this table lists those veneer grades that can be run on the given machine.

TABLE 1.3 Veneer restrictions for the dryer operation

DRYER	THICKNESS	WIDTH	LENGTH	WOOD TYPE	SPECIES
1	ALL	27",54"	4',8', and 9'	ALL	ALL
2	ALL	27",54"	4',8', and 9'	ALL	ALL
3	ALL	RW, 27", and 54"	8',9', and 10'	ALL	ALL
4	ALL	RW, 27", and 54"	8',9', 10',and 12'	ALL	ALL
5	ALL	RW, 27", and 54"	8',9', 10',and 12'	ALL	ALL
6	ALL, except 1/10" 12' heartwood	RW, 27", and 54"	8',9', 10',and 12'	ALL	ALL
7	ALL	RW, 27", and 54"	4',8', 9',and 10'	ALL	Hemlock, and noble fir

The jointer operation, on the other hand, is limited by the veneers appearance grade. Specifically, the jointer operation consists of two string machines and three edge-gluers. Since the string machines produces a less desireable joint they are not used for jointing face grade veneers. Furthermore, string machine two cannot joint veneers whose length is greater than eight feet. The edge-gluing machines, however, do produce quality joints but are also limited by veneer length. That is, machine two cannot handle 12 foot sheets and machine three can neither joint 12 foot nor 10 foot sheets. Finally, the spreader operation is limited by panel length and grade. Table 1.4 gives the length restrictions. The grade constraints are primarily imposed for cleanliness reasons. The only

TABLE 1.4 Panel length restrictions on spreader operation

SPREADER	ASSOCIATED PRESS SIZE	LENGTH LIMITATIONS
1	8′	8′ only
2	12′	9′, 10′, and 12′
3	10′	8′, 9′, and 10′
4	8′	8′ only
5	10′	8′, 9′, and 10′
6	12′	9′, 10′, and 12′

restriction requires that all duraply panels be constructed on spreader four and six. Another limitation that is unrelated to machine constraints is storage capacity. According to management, the system is limited to only six million 3/8 surface feet of dry veneer.

1.3.3 Operating Procedure

Apart from the system constraints discussed in the previous section, the plants operating procedure dictates when the machines will run. For the four stages that comprise the dry-end, the utilization of each operation will depend on the set of tasks to be scheduled. However, management policy requires that the spreader schedule be developed so that five machines are operating on day shift and swing shift, and three machines are running on graveyard shift. The same policy is not reserved for the other three operations. In fact, these operations are scheduled solely on the veneer needs created by the spreader schedule.

CHAPTER II

LITERATURE REVIEW

2.1 PLYWOOD SCHEDULING

From the volumes of literature on job scheduling, the field is vast, encompassing solution techniques for a variety of scheduling problems. Among the areas that are most frequently studied, job shop scheduling tops the list. One reason for much of this exposure is the number of manufacturing processes that can be modeled as job shops. As pointed out in chapter one, plywood manufacturing is one example. Unfortunately, the studies performed on plywood scheduling are few in number. The reason for this can primarily fall on the dimensionality imposed by this type of system. Specifically, apart from the fact that plywood facilities are modeled as job shops, the complicating twist is that each of the k operations are usually composed of m parallel and identical machines.

Although a plywood facility can be difficult to model, studies have been conducted in which algorithms were developed for scheduling plywood production. One such study conducted by Weldwood [1971] concentrates on achieving the annual operating plan through short-term scheduling. Weldwood's emphasis was not on job sequencing but on tracking the divergent factors of plywood production so that the details of the operating plan would be met. The factors he mentioned as being the demise of this plan included product mix requirements, plant capacity, and raw material limitations. He added that because two of these factors are influenced by market conditions, an effective planning heuristic should be used to ensure that the planned requirements are achieved. The approach he suggested to accomplish this goal is to formulate and solve the problem as a linear program. As far as the present study is concerned, the term scheduling has been used out of context. A more appropriate term would be short-term planning instead of short-term scheduling.

In a more applicable study, Yaptenco and Wylie [1970] viewed a plywood facility as a discrete number of components (production centers) that interact with each other only at a discrete number of points (points where inputs are received and outputs removed). Moreover, they acknowledged the problem at hand as scheduling tasks through an assembly-type production environment where the tasks must be sequenced so that the waiting time between the current and succeeding operations be kept to a minimum. Their proposed resolution of the problem is to use an appropriate time interval as a criterion for starting a succeeding operation with a portion of a task that has not yet been completed on a preceeding operation. That is, schedule the tasks so that the completion time of a given task on a preceeding operation exceeds the starting time of that same task on a succeeding operation. This is not a sufficient condition for optimizing waiting time but a necessary one. Finally, they added that "the choice of an appropriate interval or period will depend on how discrete the jobs (tasks) are, i.e., how many units make up a job and the time it takes to process a unit of a job."

The proposed modeling approach involves formulating sets of linear equations to define the inflows and outflows of each of the discrete production centers. These equations are then converted to constraints and the problem is solved using the simplex method of linear programming. The drawbacks of this model includes increased computer run time as the mill size increases and the selection of a good time interval. The problem associated with this latter drawback involves a trade-off between computer run time and the development of a detailed schedule. Specifically, Yaptenco and Wylies found that the formulated model can only be applied to an eight hour interval. Therefore, if a scheduler is interested in a more detailed schedule, this approach is not feasible.

2.2 INVENTORY CONTROL AND SCHEDULING TECHNIQUES

2.2.1 Overview of Problem

The development of a scheduling algorithm for sequencing tasks through a four stage dry-end process, as Yaptenco and Wylie have shown, is definitely a feasible undertaking. However, developing an algorithm that also ensures that a given sequence of tasks is optimal is without question a sizable problem. This does not imply that it cannot be done but the means by which this is achieved may be quite cumbersome. Even if a technique is found for optimally scheduling plywood production, it only ensures that a single measure of performance is optimized. In many cases, the true test of a production schedule is based on how well it optimizes several objective measures. As a result, a more appropriate methodology for solving this scheduling dilemma may be goal programming because of its ability to accomodate several objective parameters in a single problem. Unfortunately, by using goal programming, we still fall into the same trap as did Yaptenco and Wylie, this being computer run time. Therefore, we will not attempt to develop an algorithm that ensures optimal results but one that satisfy managements expectations.

2.2.2 Inventory Control Problem

As discussed in chapter one, the problem in formulating a schedule lies primarily with ensuring that a given sequence will time the correct quantity of veneer so that it will be available on successive operations. The scheduler must choose from among several plausible sequences that will best suit his immediate needs. These choices are provided because there is no unique way of acquiring a single veneer grade. That is, in most cases, veneer grades such as CP, C, and D grade can be developed by upgrading alternative veneer sources on either the patcher or jointer operations (or both). The final decision, though, will depend of course on the current veneer inventory as mentioned in chapter one.

Aside from the choices of how each veneer grade can be developed, the necessity to ensure that each task in the sequence will be available for further processing on successive operations is of paramount importance. This has been defined by many authors as time-phasing. To address this problem, we will avoid the current topics in scheduling literature and concentrate on a topic in inventory management, this being material requirements planning (MRP). The reason is justified by the fact that MRP recognizes the realities of demand (or veneer requirements on successive operations) existing in a manufacturing environment. That is, MRP realizes that demand does not occur at discrete points in time but occurs sporadically throughout the planning horizon. This is the basic difference that separates MRP from other inventory control techniques. The others assume constant demand and as a result do not time-phase inventory requirements over the planning period as does MRP. Time-phasing is therefore a hallmark of all MRP systems.

Time-phasing is defined by Orlicky as "capturing or developing the information on timing, so as to provide answers to the following questions."

1) When is the quantity on order due to come in, and is it a single order or are there more than one?

2) When will the demand actually have to be satisfied, and will it be at one time or is the requirement a summary figure concealing several demands spaced in time?

3) When will the stock run out?

4) When should the replenishment order be completed?

5) When should it be released?

In reference to the system under study, these questions are identical to those the scheduler must address when composing the schedule for the three operations that preceed the spreaders. Specifically, in the language of MRP, each of the end-items (panels) that comprise the master production schedule (MPS) for the spreaders must be exploded back through the three upstream operations. By this explosion process, each end-item is broken up into its components (veneer). These dependent components are then scheduled on one or more of the three upstream operations based on the planned order release for the end-item at the spreaders. The complicating feature that is exhibited in this system is that several veneer grades may exist for a single component-item. For example, figure 2.1 illustrates this situation with a branching tree diagram that is suppose to resemble the bill of material for a 3/4 AC 5 ply panel. Note that at the component level several alternative

Panel grade: 3/4" 8' AC 5 ply

Component	1 (face)	2 (back)	3 (center)	4 (xband)
Alt 1 -	1/6 54 solid	1/6 54 C	1/6 54 C	1/8 RW C
Alt 2 -	1/6 54 BP	1/6 54 D	1/6 27 C	1/8 4' C
Alt 3 -			1/6 RW C	1/8 27 C
Alt 4 -			1/6 54 D	

Figure 2.1 Panel grade construction diagram.

component-items exist. For component-item 1, either 1/6 solid or 1/6 BP veneer can be used to fill the face veneer requirement. Furthermore, component-item 4 has up to three different veneer sources. The decision of which alternative to select, however, will depend on the current inventory position of the system and possibly some processing considerations. Naturally, if an order release has been planned with 1/6 solid to be ready at the spreaders when needed, then there is no reason to consider the second alternative in component-item 1. On the other hand, if a release had not been planned then the second alternative would have to be a contender. By no means does this exhaust all of the possible situations that may exist when deciding which alternative to choose but it should give a flavor of what the scheduler must consider.

The concepts involved in material requirement planning may prove fruitful in answering the scheduling problems imposed by the three operations that precede the spreader operation. However, this approach cannot be used unless a master production schedule is first developed for the spreader operation.

2.2.3 Machine Scheduling Problem

In the eyes of the mill scheduler, the scheduling problem for the entire system is dependent on how jobs are allocated and sequenced on the spreader operation. This not only holds true for the dry-end department, but it also applies to the department that precedes (green-end) as well as the departments that follow (finish-end) the dry-end. Therefore, careful consideration is given when composing this schedule to ensure that the work load on preceding operations will be balanced and no one operation will be given a disproportionately high amount of processing that may result in a bottleneck.

Outside of the study conducted by Yaptenco and Wylie, no other research has been performed to address this problem for a job shop consisting of multiple machines in multiple operations. The vast majority of articles that vaguely resemble this system only consider multiple machines on a single operation. Among the list of authors who have studied the parallel machine problem McNaughton [1959], Root [1965], Rothkopf [1966], Baker [1974], Barnes and Brennan [1977], and Rajaraman [1975] are only a few.

To address the specific problem of scheduling the spreader operation, several methodologies may prove useful. That is, since the primary goal in scheduling this operation is to maintain continuous job throughput, we must first decide on an appropriate measure of performance that will adequately achieve our primary objective.

As described by Baldwin [1974], plywood is a composite product in which different veneer grades are glued and pressed to form panels. These panels can consist from as low as three plys of veneer to as high as seven plys. When scheduling the spreader operation, the scheduler must be aware of the number of different veneer grades that exist in each panel grade of his order file so that he will not sequence a pair of jobs that require a large number of veneer change overs. This approach allievates long set up times between jobs and ensures good job throughput.

To this end, it would seem that the best method to ensure good job throughput is to minimize the total set up times required in manufacturing a set of panel orders. Authors have classified this sort of problem as the dependent set up time problem. The objectives behind the algorithms used in solving this type of problem is to sequence jobs so as to minimize the sum of the individual set up times. That is, sequence the order file so that the unproductive waiting time associated with veneer change overs is minimized.

Two authors, Baker [1974] and King [1975], have identified this set up time problem as a form of the travelling salesman problem where the set up times in changing from one job to another are analogous to the travel cost between locations in the distribution problem context.

The methodologies provided for solving this problem include dynamic programming, branch and bound, and integer programming. A fourth method uses a near optimal heuristic similar to the closest unvisited city (CUC) approach. This latter method is only prescribed for the single machine problem but it may be adaptable to the m machine case too. The CUC algorithm works by always selecting the closest city not yet visited. In terms of the scheduling model, this rule amounts to the shortest set-up time sequencing. This procedure can easily be facilitated by setting up a square n by n matrix where the rows and columns represent the n jobs (see figure 2.2). The jobs are then sequenced by

job	1	2	3	.	.	.	N
1	-	s_{12}	s_{13}	.	.	.	s_{1N}
2	s_{21}	-	s_{23}	.	.	.	s_{2N}
3	s_{31}	s_{32}	-	.	.	.	s_{3N}
.	.	.	.	-			
.	.	.	.		-		
.	.	.	.			-	
N	s_{N1}	s_{N2}	s_{N3}				-

Figure 2.2 Set up time matrix

selecting, from the n by n elements, the ordered pairs with the smallest set-up times. For example, suppose we are given a six-job matrix (shown in figure 2.3) and were asked to find the best sequence of jobs using this algorithm. To sequence these six jobs, first select the pair of jobs with the smallest set-up time s_{ij}. The initial starting point, i, is referred to as the origin. In this case, the origin will be job four and the sequence with the smallest set-up time is either s_{46} or s_{43}. If we choose s_{43} as our initial sequence, the next job in the tour is determined by excluding job four and selecting the smallest set-up time in row three (s_{3j}). From the matrix, s_{35} is best. This procedure is continued until all jobs have been eliminated from the matrix. In our example the best sequence starting with s_{43} is 4-3-5-6-2-1 with a total set-up time of 18. On the other hand, had we started with s_{46} the sequence would have been 4-5-6-2-1-3 with a total set-up time of 22.

job	1	2	3	4	5	6
1	-	4	8	6	8	7
2	5	-	7	11	13	4
3	11	6	-	8	4	12
4	5	7	2	-	2	9
5	10	9	7	5	-	4
6	8	3	13	8	7	-

Figure 2.3 Set up time matrix example

The example presented above is one variation of the CUC algorithm in which the origin was selected arbitrarily and the set-up times were absolute. Another variation involves the look ahead feature that permits a closest unvisited pair of cities to be added to the tour. Under this variation the origin is chosen arbitrarily. Then, instead of examining the path from the origin to (n-1) other cities this method examines the path from the origin to (n-1) times (n-2) ordered pairs of unvisited cities and adds to the tour the pair associated with the minimum distance (in our case minimum set-up time). Because of these variation the CUC algorithm has eight possible versions as is indicated in table 2.1.

TABLE 2.1 CUC algorithms

Algorithm	Variation
1	Abs. set up time; no look-ahead; starting job arbitrarily
2	Abs. set up time; no look-ahead; all jobs chosen as origin
3	Abs. set up time; look-ahead ; starting job arbitrarily
4	Abs. set up time; look-ahead ; all jobs chosen as origin
5	Rel. set up time; no look-ahead; starting job arbitrarily
6	Rel. set up time; no look-ahead; all jobs chosen as origin
7	Rel. set up time; look-ahead ; starting job arbitrarily
8	Rel. set up time; look-ahead ; all jobs chosen as origin

The drawback of using the CUC algorithm or any of the other methods used in solving the dependent set up time problem for the spreader operation is that there is no certainty that a given schedule will ensure that the processing on the three upstream operations will be balanced. The answer to this will depend first on how each veneer component will be developed and secondly on the available machine capacity on each of the three upstream operations. For this study, it will be assumed that this machine capacity will be adequate to accomodate any schedule developed for the spreader operation.

CHAPTER III

MODELING APPROACH

3.1 OVERVIEW OF SCHEDULING APPROACH

From the preceding discussion, it should be apparent that a plywood facility is a difficult system to schedule. Much of this difficulty is due in part by the fact that the scheduler must sequence tasks in a system composed of four operations where each operation consists of m parallel processors (machines). The problem does not lie in sequencing the individual operations because this can be addressed using current scheduling algorithms. It lies in determining how each veneer grade will be developed on the three upstream operations as well as the assurance that these desired grades will be present in inventory when needed at the spreaders.

As presented in chapter two, the literature on this subject is sparse. This may be attributed to the fact that the dimensionality of the problem when dealing with large and complex systems does not lend itself well to optimal solutions. However, with the growing popularity of decision support systems it would seem that the attention is straying from these approaches that provide optimal solutions to those that only provide good solutions. Managers are finally recognizing the fact that the cost of their time out weighs the benefits derived from an optimal solution. Hence, the answer to the immediate problem of scheduling tasks in the system under question seems obvious. That is, develop a scheduling program that does not focus on the attainment of an optimal solution, but one that provides good results at a substantial savings in procurement and computer run-time.

The scheduling program developed in this study to accomplish this objective does so by emulating the process used by the actual scheduler. That is, since the job schedule for the spreader operation is the single most important factor affecting how the up stream operations (jointer, patcher, and dryer) can be scheduled, this operation is evaluated first followed by the three remaining operations. To support this approach, the program is broken down into three distinct phases (as illustrated in fig. 3.1) where each phase resembles the steps the scheduler performs in procuring a viable job schedule.

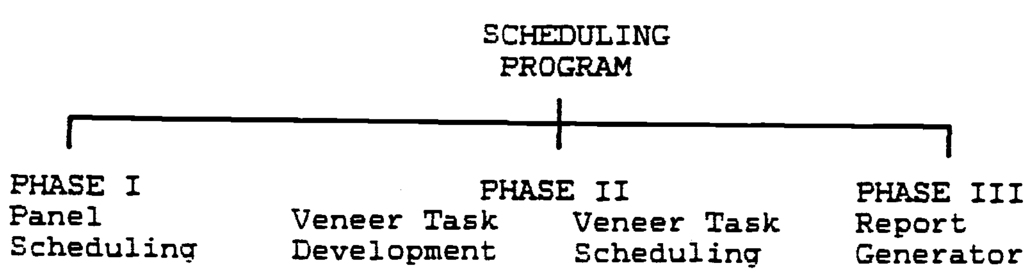


Figure 3.1 Program breakdown

The philosophy used in designing this program concentrates mainly on computer-assisted scheduling in contrast to computerized scheduling. The approach attempts to remove mill personnel from the physical chore of job scheduling and places this responsibility on the computer. Though, this should not be construed that the human has been totally displaced from this activity. In fact, the human still plays an integral role in the development of each schedule produced by this package. His role includes

1) developing the list of panel orders (TASK.DAT) that will be scheduled on the spreader operation,

2) developing the initial dry inventory (DRY.DAT),

3) formulating the idle machine policy for the spreader operation (MIDLE.DAT),

4) formulating the constraints on task assignments to the machines of each operation (GLPOLI.DAT, JPOLI.DAT, DPOLI.DAT),

5) formulating the alternative strategies for procuring a given veneer grade through other veneer grade sources (VENDEV.DAT), and

6) creating green inventory in phase two.

Of these responsibilities, the first five can be performed before the package is actually run. However, the sixth requires the user's constant attention during the running of the program. Specifically, phase two has been designed so that it interactively prompts the user for the green veneer grade that will be used to procure a dry veneer order (set up by the panel orders scheduled on the spreader operation). This is done to ensure that each dry veneer order will be constructed from a suitable green inventory source. The term "suitable" implies that the source is currently available on the log yard and it is the best green veneer grade for developing the desired veneer.

As outlined above, the program is composed of three phases. The first two phases represent the actual scheduling program. Their primary objective is to generate the sequence files, SEQ1.DAT, SEQ2.DAT, SEQ3.DAT, and SEQ4.DAT, for each of the respective operations that will be used in phase three to generate the formatted machine schedules illustrated in appendix C.

Phase one (see figure 3.2) initiates the package by generating the spreader schedule (SEQ1.DAT) using the set of user defined panel orders from the file, TASK.DAT. After reading these orders, the program first allocates them among the m machines of the spreader operation according to the machine restriction file, GLPOLI.DAT. This file is one of many policy files that the user can employ to designate the constraints he wishes to impose on the current schedule. Once the orders have been allocated to a machine, they are then sequenced using the dependent set up time heuristic

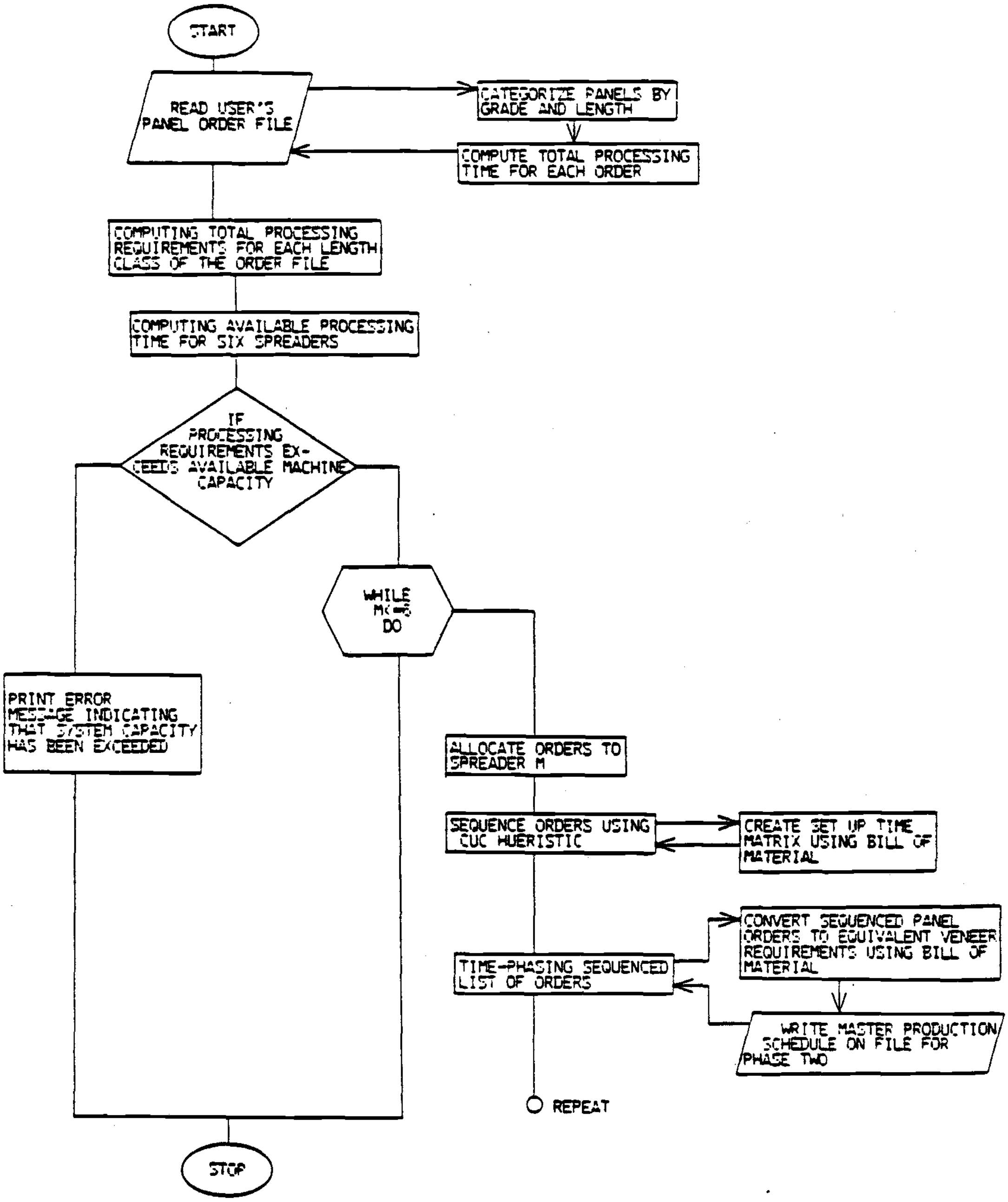


Figure 3.2 Phase one flowchart

discussed in chapter two. In the process of sequencing jobs on the spreader operation, a time phased list of veneer orders (master production schedule) is created. This list is maintained in the file, PAT.DAT, so that it may be used in phase two where each veneer grade from this list can be filled.

In terms of the scheduling problem at hand, phase two (see figure 3.3) is the most critical phase because it ensures that the schedule developed for the spreader operation will be achieved with the desired veneer grades. This is accomplished using a veneer development policy , VENDEV.DAT, which outlines the alternative strategies that may be used to develop each of the desired grades.

The algorithm works on an order by order basis. That is, the orders of the master production schedule are processed one at a time by checking a linked inventory list. Initially, this linked list is checked for the desired grade specified by the order. If this grade is found, then the needed quantity is removed, otherwise, an alternative development strategy is evaluated. This alternative strategy is obtained from the veneer development policy which designates a prioritized list of alternative sources (veneer grades) for satisfying the current order. In most cases, these alternative sources are lower grade veneers

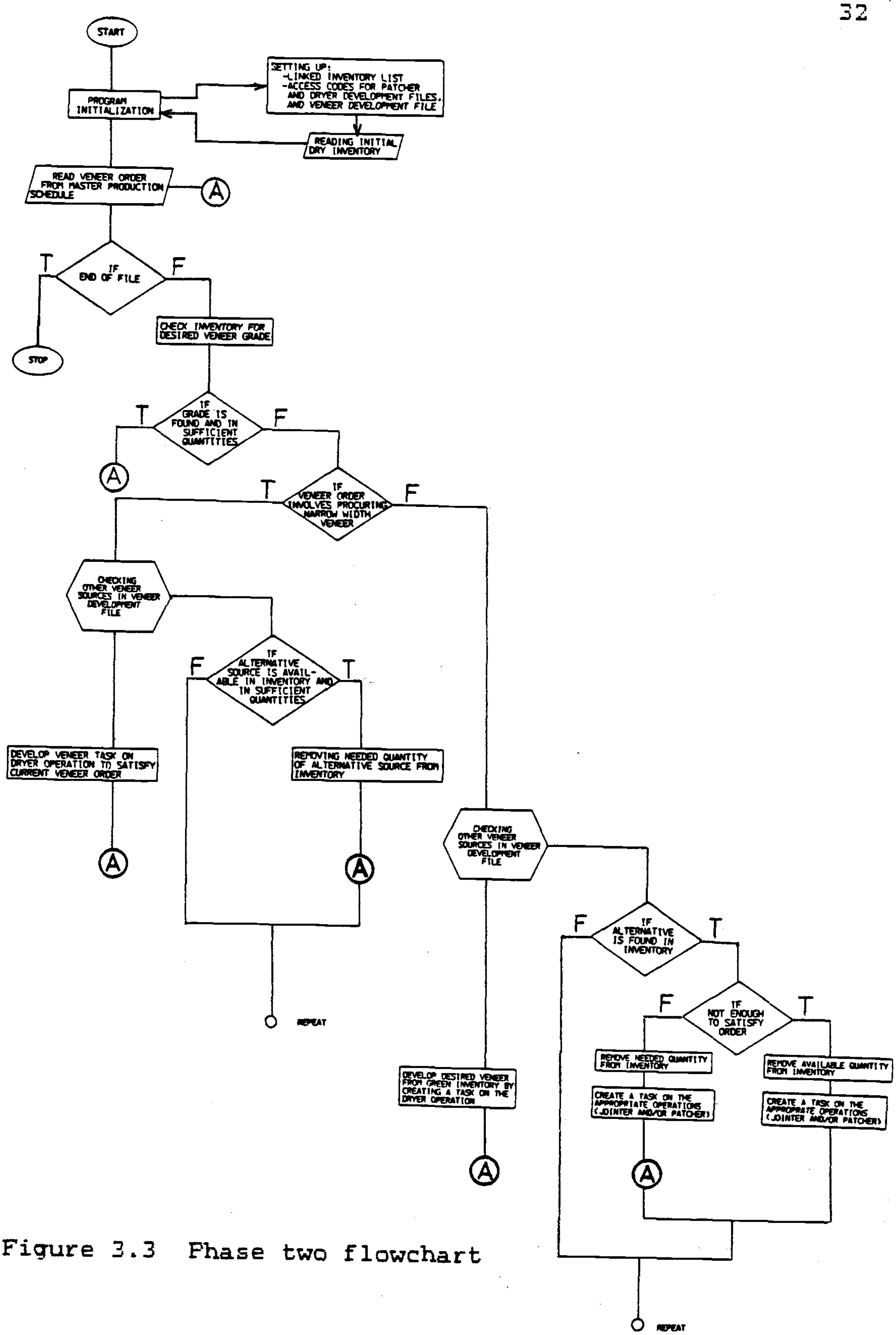


Figure 3.3 Phase two flowchart

that require upgrading on either the patcher or jointer operation (or both) before they can fill the current order.

The evaluation of the alternative sources is an iterative process in which the linked inventory list is checked for each source (veneer grade). If the alternative veneer grade is found, then the quantity is removed and a task is created on the appropriate operation that will up grade this veneer to the desired grade. However, if the grade is not located, then the next alternative source is checked until either the order has been filled or the veneer development file has been exhausted. If the latter occurs, then the desired veneer grade is obtained directly by creating a task on the dryer operation using a green veneer source specified by the user. This step is reserved for last to ensure that every possible means is exploited to develop the desired veneer from current dry inventory.

Once it has been determined that a task is required on one of the three upstream operations, the job is immediately scheduled on one of the m machines of that operation. When scheduling, the machines are prioritized based on the total amount of processing it has been currently assigned so that the machine with the smallest amount of processing will be considered first when making a job assignment. Though it should be noted that before an assignment is actually made, the task is evaluated using the machine restriction policy to determine whether the task can be allocated. If it can, the job is inserted in the schedule for that machine at a point that guarantees its availibility on either the spreader operation, or any other operation for which this task is intended. However, if the task cannot be assigned, the next machine with the smallest total processing assigned to it is checked. This procedure is repeated until an assignment is made.

The guarantee of a good solution using this scheduling package is only ensured if the user sets up his policies realistically. That is, the policies must reflect the current manufacturing environment as well as the user's wishes and expectations. If these are not considered, then erroreous results could occur when implementing this package.

3.2 MODELING ASSUMPTIONS

In designing this scheduling package, several assumptions were made regarding the system being modeled. Most of the assumptions are typical of those the scheduler makes; however, others were made because of the limitations imposed on the scope of this project. These assumptions are as follows:

1) All of the jobs to be scheduled on the spreader operation will be known at the outset.

2) Breakdowns or repairs of any machines will not occur during the planning period.

3) Production capacity of the green-end department is adequate to handle the production requirements of the dry-end department.

4) Production capacity of the three upstream operations are adequate to accomodate the requirements of the spreader operation.

5) Processing rates are also given at the outset and are constant regardless of the ordering of processing.

6) Processing rates will not include set up times for the spreader operation.

7) Production is performed in batches and a batch cannot be split among operations.

In the next chapter, a more detailed accounting of the actual program is presented. Each section elaborates on one of the three phases used in developing the dry-end schedule. To assist the reader, the FORTRAN subprograms are provided in appendix A.

CHAPTER IV

DESCRIPTION OF SCHEDULING PROGRAM

4.1 PHASE I: SPREADER SCHEDULING

4.1.1 Introduction

As presented in the branching tree diagram of figure 4.1, phase one is a 21 subroutine package that acts as the driver for this scheduling program. Its primary function, as stated in chapter three, is to sequence the spreader operation with the user defined tasks in a way that minimizes total waiting time. Another function of this phase is to generate a master production schedule from the spreader schedule so that jobs can be assigned to the three upstream operations in phase two.

4.1.2 Input of the Panel Order File

Phase one starts by first reading in the set of user defined orders from the file TASK.DAT in subroutine CODINP. Each order is identified with a six digit code followed by the quantity desired as illustrated in figure 4.2. As each order is read, it is translated in subroutine INCONS using the panel description file, PANEL.DAT (see appendix B). This translation involves computing the total processing time for the order, and setting up the bill of material (veneer grades) that will be used to assemble each panel.

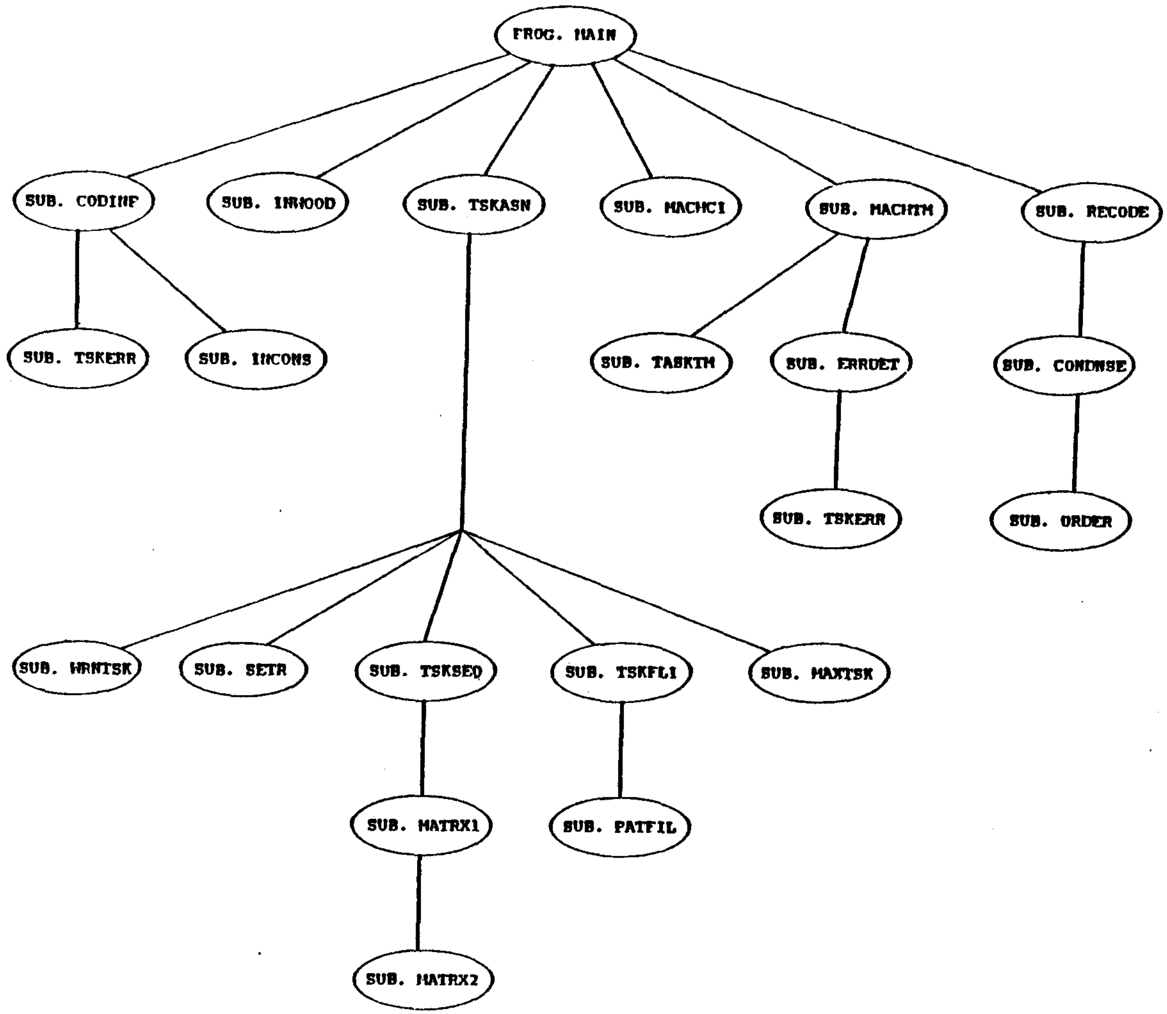
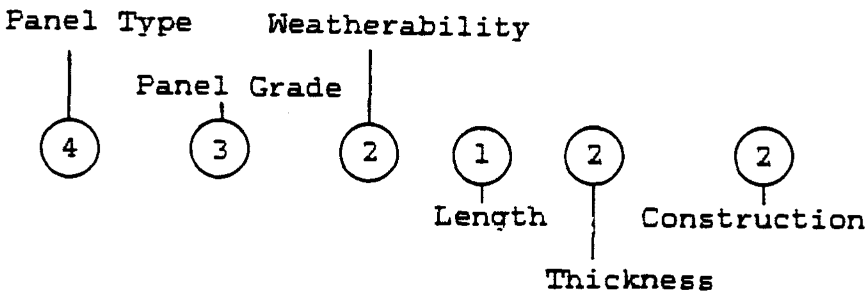


Figure 4.1 Phase one tree diagram

Without the bill of material neither the set up time matrix (used for sequencing jobs) nor the master production schedule could be created. In addition to translating each panel order, subroutine INCONS also designates the panel



Panel Type	Grade	Weatherability
1-Siding	1-8	1-Exterior
2-Sanded		2-Interior
3-Marine		
4-Duraply		
5-Sheathing		

Length	Thickness	Construction
1-8′	0-5/16	1-4
2-9′	1-1/4	
3-10′	2-3/8	
4-12	3-1/2	
	4-5/8	
	5-3/4	
	6-1 0/0	
	7-1 1/8	
	8-1 1/4	
	9-1 1/2	

Figure 4.2 Panel code description

grade and length of each order so that the task used in constructing each order will be properly assigned to one of the six spreaders in the allocation subroutine, TSKASN.

4.1.3 Capacity Check

After the order file has been read and translated in subroutine INCONS, a capacity check is made. That is, the total processing times for the four length categories (from the users order file) are computed in subroutine TASKTM and then compared with the available processing time (computed in subroutine MACHTM) for each of the six spreaders. If the current set of orders is found to exceed the available machine capacity, then an error message is written informing the user that either he must modify his current order file or reformulate the idle machine policy for the spreader operation (in file MIDLE.DAT). If, on the other hand, the processing time does not exceed the available time, then the order file can be sequenced on the spreaders. This is accomplished using two subroutines, TSKASN and TSKSEQ. Each routine addresses the two functions of scheduling. The first subroutine, TSKASN, performs the task allocation activity whereas the second subroutine, TSKSEQ, performs the sequencing activity.

4.1.4 Task Allocation

In the allocation subroutine, TSKASN, the orders are distributed among the six machines using the panel type (grade and lengths) array, constructed earlier in subroutine, INCONS, and a machine restriction policy (defined in the file GLPOLI.DAT). As discussed in chapter three, this machine restriction policy (see table 4.1) allows the user to set constraints on how panels can be assigned to the six machines of the spreader operation. The policy, specifically, prioritizes the panel grades as well as lengths on each machine so that only the highest grade -length combination will be considered first on any given machine. For example, when considering machine six the program will attempt to fill this machine with 12 foot duraply ahead of any other grade-length combination. If there is no 12 foot duraply available in the current set of orders, then the next highest item on the list will be checked, in this case, 10 foot duraply. This procedure continues until either an order with the correct grade-length combination has been found or the policy list has been exhausted. If the correct panel is found, then the program will continue filling the current machine with this panel type until either the available processing time is exceeded or the supply of the correct panel has run out, in which case, the program will check the current order set for the next highest item on the priority list.

4.1.5 Task Sequencing

After a machine has been filled, the set of tasks

Table 4.1 Glue spreader restriction file

Machine		Priority 1	2	3	4	5	6	7	8	9	10	11	12
1	Grade	Siding	Marine	Sanded	Sheathing								
	Length	8'	8'	8'	8'								
2	Grade	Siding	Siding	Siding	Siding	Marine	Marine	Marine	Marine	Sanded	Sanded	Sanded	Sanded
	Length	9'	10'	12'	8'	9'	10'	12'	8'	9'	10'	12'	8'
3	Grade	Sanded	Sanded	Sanded	Marine	Marine	Marine	Siding	Siding	Siding	Sheathing	Sheathing	Sheathing
	Length	10'	9'	8'	10'	9'	8'	10'	9'	8'	10'	9'	8'
4	Grade	Duraply	Marine	Sanded	Siding	Sheathing							
	Length	8'	8'	8'	8'	8'							
5	Grade	Sanded	Sanded	Sanded	Marine	Marine	Marine	Siding	Siding	Siding	Sheathing	Sheathing	Sheathing
	Length	10'	9'	8'	10'	9'	8'	10'	9'	8'	10'	9'	8'
6	Grade	Duraply	Duraply	Duraply	Duraply	Marine	Marine	Marine	Marine	Sanded	Sanded	Sanded	Sanded
	Length	12'	10'	9'	8'	12'	10'	9'	8'	12'	10'	9'	8'

assigned to this machine are then ready to be sequenced in subroutine TSKSEQ. This is done using the CUC sequencing algorithm discussed in chapter two where the set-up time matrix is created first using the bill of materials for each of the assigned tasks. Specifically, this matrix is constructed using subroutines MATRX1 and MATRX2, by evaluating the setup times for every possible ordered task pair (panel jobs) assigned to the current machine. The set up time is computed by determining the number of veneer grade changes that occur when changing over from one panel grade to another. As an example, the number of veneer changeovers that occur when changing from an 8' 5/8" AC panel to an 8' 3/4" AB panel will depend on the grade, thickness, and length of the face, back, center, and crossband plys of each panel. The grade, thickness, and length breakdown for these panels are given in figure 4.3. As is shown, two veneer changeovers occur when pairing these two grades together.

Panel	Face			Back			Center			Xbnd	
	grd	thk	len	grd	thk	len	grd	thk	len	grd	thk
8 5/8 AC-	A	1/10	8	C	1/10	8	C	1/6	8	C	1/6
8 3/4 AB-	A	1/10	8	B	1/10	8	C	1/6	8	C	7/32

Figure 4.3 Bill of material for 8' 5/8" AC and 8' 3/4" AB

Therefore, if these grades had actually been included in the set of tasks assigned to the current machine, the set up time would be 60 seconds, allowing 30 seconds for each change over. This procedure is repeated until all of the possible order task pairs assigned to the current machine have been given a set up time.

Once this set up time matrix has been ascertained, the tasks are then ready to be sequenced using the all-origins version of the CUC algorithm. As mentioned above, the actual sequencing is performed in subroutine TSKSEQ where only the first P tasks assigned to the current machine are used as origins. This ensures that the panels with the highest panel grade and length priority (as determined by the machine restriction policy) are scheduled ahead of all other tasks assigned to this machine.

4.1.6 Development of Master Production Schedule and Sequence File

After a machine is scheduled, the ordered jobs are then time-phased in subroutine TSKFIL so that each job will be associated with its own start time. This is performed so that the master production schedule (stored in the file PAT.DAT) will be constructed and the sequence file, SEQ1.DAT, will be created for the report generation phase.

4.2 PHASE II: VENEER TASK SCHEDULING

4.2.1 Introduction

As presented in an earlier discussion, phase two is concerned with the fulfillment of veneer orders created by sequencing jobs (panels) in phase one. In some respects, this phase resembles a modified material requirements planning (MRP) package because of the way orders are filled by creating tasks among the three upstream operations (patcher, jointer, and dryer). However, the main distinction between this algorithm and most MRP packages is that several options are possible in satisfying a parent-item.

The resolution of this problem is established by allowing the user to enumerate his preferrences for developing every possible veneer grade currently recognized by the modeled system. The file that stores this information is called VENDEV.DAT and hereafter is referred to as the veneer development file. This file prioritizes the list of options that the user designates for developing each veneer grade. These options include the grades as well as widths of each of the alternative veneer sources. The intent of this file is to eliminate the question of how each veneer order is to be procured by setting up a prioritized list of alternative veneer sources that may be used if the desired veneer grade is not satisfied with current inventory. The most current version of this file is listed in appendix B.

The primary objective of this phase is to assign and schedule tasks on the three upstream operations so that the orders of the master production schedule are filled. However, these orders must be filled in a way that attempts to deplete current in-process inventories. This criterion must be considered to prevent dry inventory from exceeding the space capacity of the system. Hence, the program will always attempt to satisfy orders of the master production schedule in a way that diminishes the current dry inventory instead of increasing it.

4.2.2 Task Development

As is illustrated in figure 4.4, phase two is partitioned into two separate programs. The first program performs the task development function in which the veneer orders are read from the master production schedule and are filled either by acquiring the desired veneer directly from the linked inventory list or by creating a task on one of the three upstream operations. If a task is needed, then the second program is used to schedule this task on one of

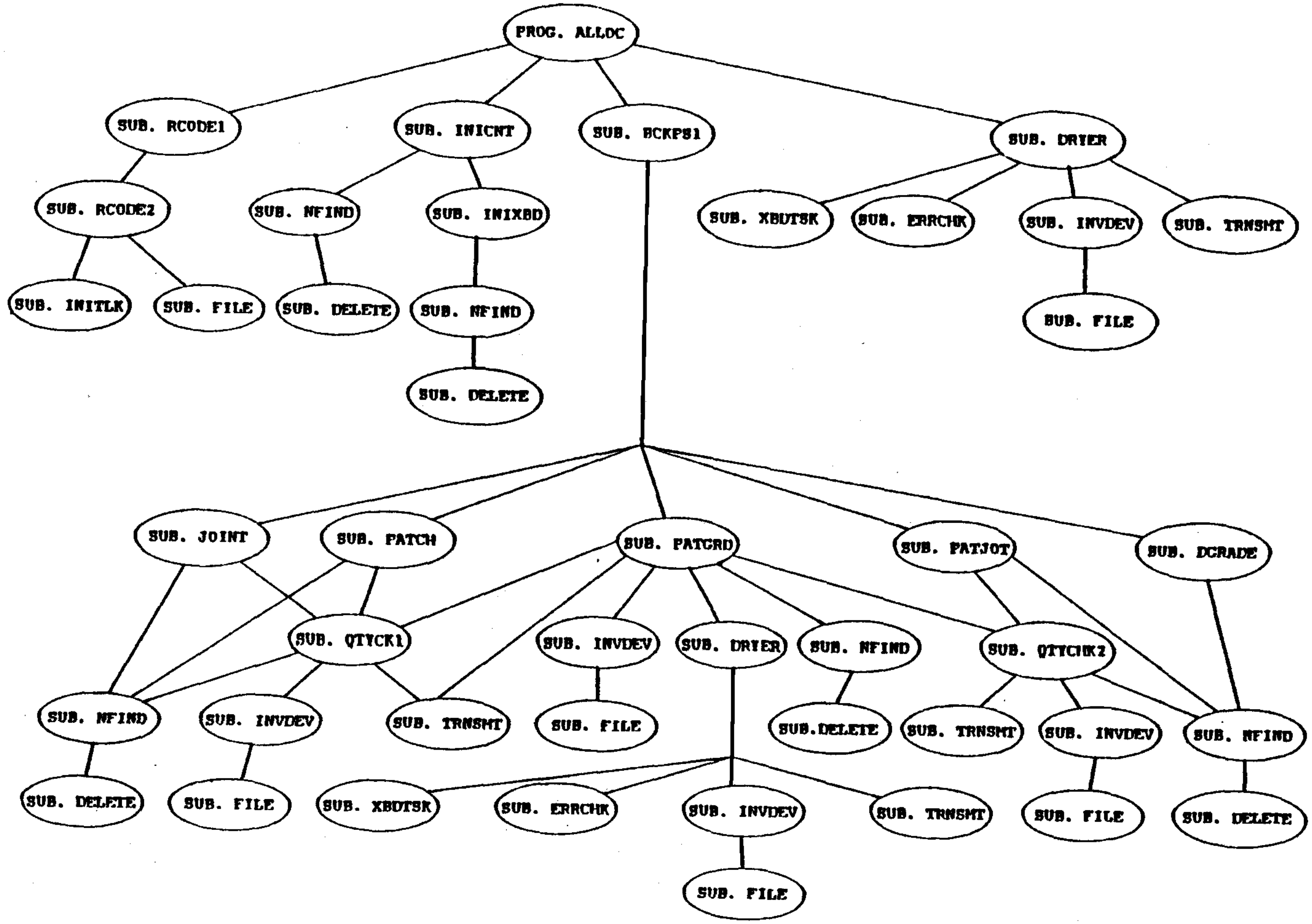
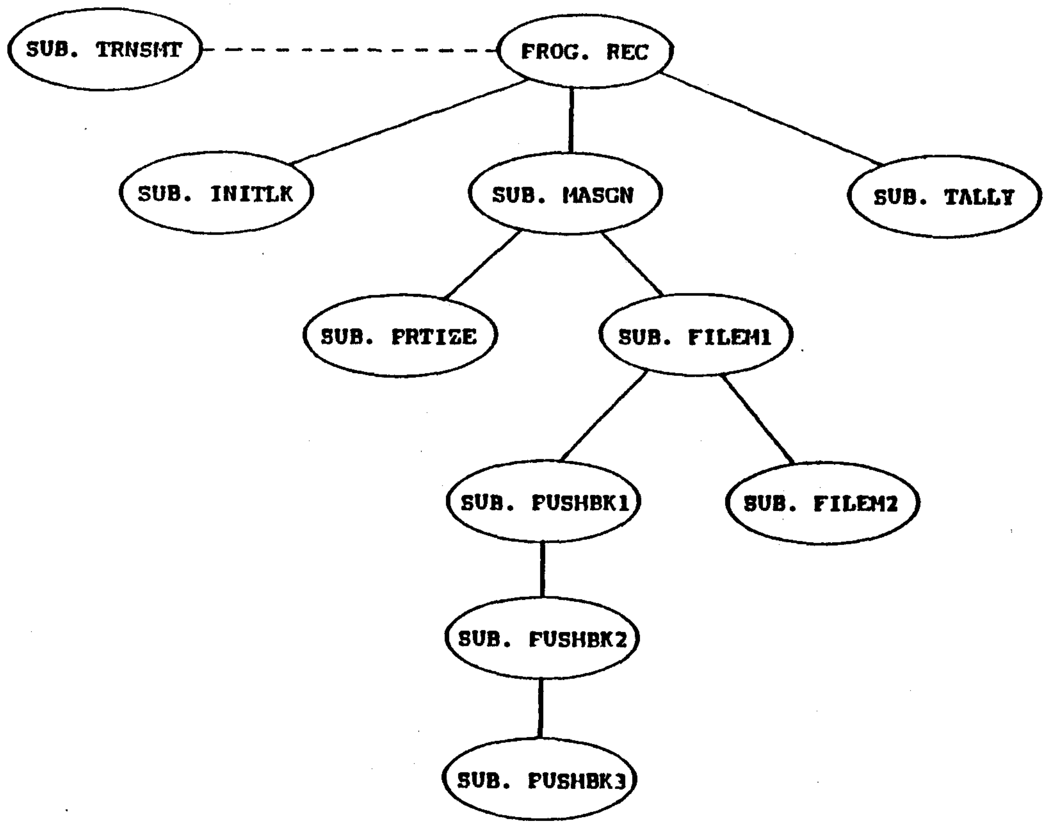


Figure 4.4 Phase two tree diagram: task development program

Figure 4.4 (cont'd) scheduling program



the m machines of the required operation. The reason for partitioning this phase is primarily due to the limitations imposed by computer memory. Since the combined memory requirement for each program exceeds the 64k limit, a single program could not be run without overstepping its bound.

As mentioned above, the first program initiates phase two by reading a veneer order from the master production schedule in program ALLOC. Thereafter, subroutine INICNT is called to make the intial check of the linked inventory list for the desired veneer. At this point, the program makes a distinction between order types. If the order type consists of procuring narrow width veneer (RW or 27"), then subroutine INIXBD is called to check inventory, otherwise, if the order involves full width veneer (54"), then the check of inventory is made directly in subroutine INICNT.

The linked inventory format used in this phase to accommodate the needs of the master production schedule was tailored after the linked list array used in the simulation language, SIMLIB, Kelton and Law [1981]. Each list represent a veneer grade and the attributes of each entry contained in a list include the quantity stored, the time the entry was placed in inventory, and the original task (number) used in creating this inventory. The second attribute is retained so that the entries will not be accessed prematurely. The third attribute is necessary for scheduling purposes. Its importance will become more apparent when the scheduling approach of the second program is discussed.

The actual subprograms used in removing and storing dry inventory are nearly identical to those used by SIMLIB. These subprograms include INITLK, FILE, NFIND, and DELETE. The first routine, INITLK, is the initialization program. This subroutine initializes the predecessor and successor pointers and zeros out the head and tail pointers for each list. The remaining subroutines facilitate the storage and retrieval functions. Subroutine FILE is identical to that used in SIMLIB with a few minor modification. It stores veneer on a largest storage time first priority. The other two routines, although contrived for the purposes of this package, perform the retrieval function. Subroutine NFIND searches a list for the entry whose storage time is either equal to or less than the current time. If one is found, it is removed using subroutine DELETE in which the entry is placed in the list of available space. Though, if an entry is not found or the current veneer order is filled without depleting the source, the entry is not removed from the linked inventory array.

When searching an inventory list for veneer that will satisfy the current order, each entry of that list is checked. If the order is not filled, then another means is employed to satisfy the current order. However, the method by which this is accomplished will vary according to the order type. For narrow width veneer orders, however, this does not hold because the processing requirements are limited to only the dryer operation. Conversely, methods for procuring veneer for full width orders do vary. In fact, full width orders could conceivably require processing on one, two, or all three of the upstream operations depending on the current inventory position for a given order grade.

Since narrow width orders do not impose a significant problem in terms of processing, only a brief outline of how these orders are filled will be given.

As stated earlier, an initial check of the linked inventory list is made for both order types in subroutine INICNT. However, for narrow width order, the actual call to subroutine NFIND is not made in subroutine INICNT, but subroutine INIXBD. Both subroutines, though, set up the iterative process in which each alternative veneer source of the veneer development file is considered as a possible candidate for filling the current order. If an alternative veneer source is available, then the current order is satisfied. However, if it is not available (or it is available, but not in sufficient quantities), then the next alternative source of the development file is checked until either the order is filled or the development file has been exhausted, in which case, the order is procured by calling subroutine DRYER to develop the desired grade from green inventory.

The complexity of how full width orders are procured is primarily a function of how the user defines the veneer development file. For example, an illustration of how 1/6 C grade veneer can be developed is given is figure 4.5. Note that eight

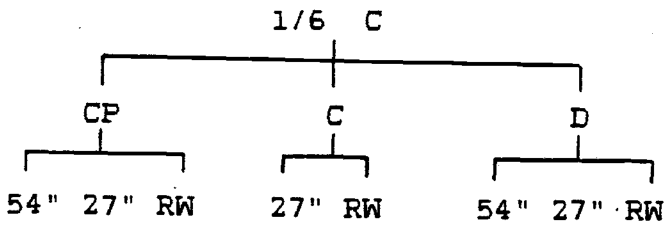


Figure 4.5 Development chart for 1/6 C grade veneer

sources are available to satisfy this grade. However, the processing requirements of each source may vary. In the cases where either 27" or RW veneers are used, a jointer task must be created. Furthermore, if D grade veneer is used, in addition to a jointer task a patcher task must also be created. As a result of these variable processing requirements, the program has been equipped to handle the following processing options:

1) Drying only
2) Patching only
3) Jointing only
4) Jointing and patching
5) Drying, jointing, and patching
6) No processing or downgrading

The central routine used to support these options is subroutine BCKPS1. This subroutine does not actually create the tasks to satisfy each order but explores the various options prescribed in the veneer development file. Its primary function is to ensure that the appropriate subroutine is called (based on the alternative veneer grade specified by the veneer development file) so that the correct task will be constructed.

The routines that subroutine BCKPS1 calls to create these tasks include:

1) Subrountine PATCH
2) Subroutine JOINT
3) Subroutine PATJOT
4) Subroutine PATGRD
5) Subroutine DGRADE

For the situation where the alternative veneer source requires only patching to upgrade it to the desired grade, subroutine PATCH is called. Similarly, if the veneer width of the alternative veneer source is less than 54", then a jointer task is needed and subroutine JOINT is called. In either case, these routines are only used to check the linked inventory list for the alternative source. If the desired grade is found, then a task is created in subroutine QTYCK1. However, if the quantity found in inventory is not enough to satisfy the current order, then subroutine QTYCK1 continues the search (of inventory) by updating the time that the alternative veneer source is needed by the amount of processing made available by the veneer previously found. This heuristic is illustrated in figure 4.6 with the use of a time line. Note that time P+S is the time the desired grade is needed at the spreader operation and time P is the start time (on either the jointer or patcher operation) for processing the alternative veneer source used to develop the desired grade. If, after initially checking the linked inventory list and finding only N time units of the alternative source in inventory to be available at or before time P, we can add this amount to P and recheck inventory again, however, this time looking for dry inventory that is available at or before time P+N. This process is repeated until the entire S time units is filled or we exhaust the alternative veneer source in dry inventory. If inventory is depleted before we fill the current order, then the next alternative veneer source is checked. This alternative source, though, may not be governed by the same processing options (Patch only or Joint only). If it is, then the

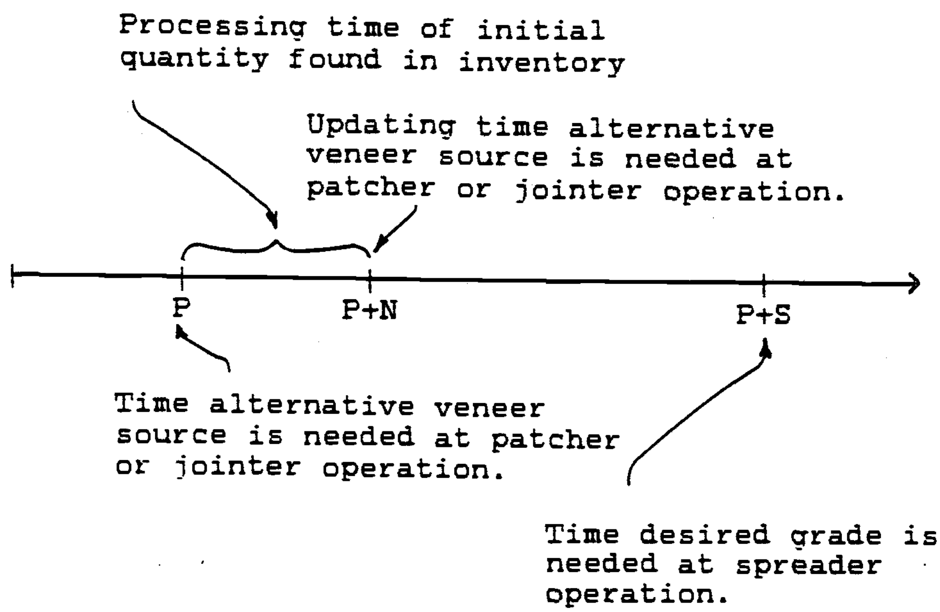


Figure 4.6 Veneer acquisition example

above steps are repeated. However, if this new alternative requires processing on both the patcher and jointer operations, then subroutine PATJOT is called. Conversely, if the alternative requires no processing, then subroutine DGRADE is called.

Subroutine PATJOT is nearly equivalent to both the patcher task development subroutine, PATCH, and the jointer task development subroutine, JOINT. Specifically this routine attempts to locate the desired narrow veneer in the linked inventory list. If found, then a patcher as well as jointer task are created in subroutine QTYCK2. If the quantity found does not satisfy the current order, then subroutine QTYCK2 continues the search in the same manner subroutine QTYCK1 did.

Subroutine DGRADE, on the other hand, is used to fill orders for alternative veneer grades that do not require processing. This routine is analogous to subroutine INICNT in that it simply checks the linked inventory list for the alternative source. If an item is located that satisfies the order, then the needed quantity is removed; otherwise, the next alternative source is considered.

The previous discussion has outlined the methods by which orders are fulfilled using the veneer development file. All of the veneer grades that can be satisfied using this approach are those that can be obtained directly from the dryer operation. Therefore, if all of the alternative veneer sources are exhausted, then the desired grade can be obtained by drying green inventory at the dryer operation. What about those grades that cannot be developed directly from a green inventory source? For the system being modeled, these grades include A, B, CP, and M grade. In each case, the grades are developed from an intermediate source. For example, A and B grades are created by patching

BP. Furthermore, M grade is produced by patching either DCLR or ECLR. As a result of these processing stipulations, these four grades are developed using subroutine PATGRD.

Like the other subroutines used in creating veneer tasks for filling a full width veneer order, subroutine PATGRD is called from subroutine BCKPS1. Furthermore, the means by which PATGRD constructs veneer task is nearly identical to that used in the other subroutines called by BCKPS1. Specifically, PATGRD is divided into two distinct parts; one section for developing tasks only on the patcher operation and the other for developing tasks on both the patcher as well as the jointer operations. This division was incorporated to allow for the possibility that a veneer order may only be developed from a narrow width intermediate veneer source. Like subroutines PATCH and JOINT, subroutine PATGRD initially checks the linked inventory list for the intermediate grade. If any veneer is found, it is used to construct a patcher task (and passibly a jointer task); otherwise, the intermediate grade is obtained directly from the dryer operation. This normally results in the creation of at least two tasks, one for drying the intermediate grade and the other for patching it. If the intermediate grade is less that 54", a jointer task is also created.

As mentioned above, subroutine BCKPS1 is the pivotal routine that ensures that every attempt is made to fill the current veneer order with avialable dry inventory. This eliminates the need of creating new inventory from green stock. However, if after exhausting every possible option the order still has not been satisfied, then the order is procured by creating new inventory using subroutine DRYER.

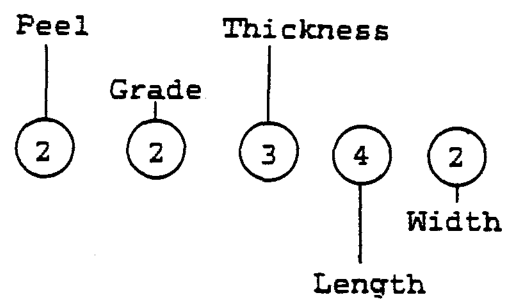
Like the subprograms just discussed, subroutine DRYER's primary function is to satisfy the current order by generating the desired grade from green inventory. However, subroutine DRYER does not access the required green veneer from a linked storage array like the other subroutines do. Instead, it interactively prompts the user for the necessary veneer as is illustrated in figure 4.7. This approach is

ORDER TYPE: CENTER	ORDER TYPE: XBND
VENEER GRADE: D	VENEER GRADE: SOL
THICKNESS: 1/12	THICKNESS: 1/8
LENGTH: 8	LENGTH: 10
WIDTH: 54	WIDTH: RW
ORDER QUANTITY: 2.50	ORDER QUANTITY: 1.25
	ACTUAL VENEER REQTS
FULL VENEER CODE:	USING 4: 5.34
	8: 3.24
	12: 1.89
	XBND VENEER CODE:

Figure 4.7 Users green inventory prompt

used to provide the user with the luxury of developing a green veneer order list that more realistically reflects the current manufacturing philosophy and climate.

The actual subprogram prompts the user for a five digit green inventory (see figure 4.8) code. To assist the



Peel	Grade	Thickness	Length	Width
1-split	1-1P	1-7/32	1-8′	1-54
2-reg.	2-3P	2-1/6	2-9′	2-27
	3-SM	3-1/10	3-10′	3-RW
	4-2M	4-1/12	4-12′	
	5-3M		5-4′	
	6-PC			

Figure 4.8 Green veneer code description

user in choosing the appropriate code from the green veneer file DRYDEV.DAT, the prompt includes the type (face, back, center, or xbnd), grade, thickness, length, width, and quantity of the veneer order. If the order type involves a narrow width veneer (as indicated by xbnd), the user is also provided with the equivalent veneer quantities for the 4′, 8′, and 12′ length classes. After the five digit code has been selected and input, an error check is made to see if the code is viable. If it is, the user will then be asked whether he would like to override the system and specify his own veneer quantity for the dryer task being created. If it is not viable, the program will write an error message and prompt the user again with the same order. Likewise, if the code is viable, yet he does not specify enough veneer to satisfy the current order the program will ask for more until the order is filled.

The drawback of using subroutine DRYER is that it tends to counter our objective of minimizing in-process inventories. Outside of the patching operation, the dryers are the only other processing center where a single grade input (green veneer) can (and most likely will) result in a multi-grade (dry veneer) output. The only distinction between these two operations in terms of our objective is that the patcher will not increase dry inventory. Instead, it only changes the grade make-up of dry inventory. However, the dryer operation will. For this reason, subroutine DRYER is called only after every other source has been explored.

To facilitate these sorts of changes and/or additions to dry invertory, both subroutines PATCH and DRYER utilize a development file to determine the grades that result from a single grade input. For subroutine PATCH, PATDEV.DAT is used to evaluate these grades. For subroutine DRYER, DRYDEV.DAT is used. In either case, the actual determination and updating of dry inventory is performed in subroutine INVDEV. This routine does not directly input the newly generated dry inventory into the linked storage array, but it sets up the transfer array and calls subroutine FILE so that the new inventory will be stored correctly.

To this point, we have discussed in some detail the attributes of phase two and how it fills orders (of the master production schedule) by creating tasks on the three operations that preceed the spreaders. But, how are these tasks scheduled in the second program of this phase?

4.2.3 Task Scheduling

The actual scheduling algorithm for sequencing newly created tasks on a machine of the appropriate operation is quite simple. After a task is created in program one, the attributes of this task are transmitted via a system directive to the scheduling program, REC. The list of attributes sent include:

1) the operation the task is to be scheduled on,
2) the total required processing time for the task,
3) the production rate for the task on the assigned operation,

4) the veneer grade, thickness, length, and width of the task,

5) the completion time of the task, and

6) the raw material source code for the task.

Upon receipt of these attributes, program REC calls the machine restriction subroutine, MASGN, to determine which machine the task can be assigned to without violating the machine restriction policy prescribed by the user (in the files JPOLI.DAT, and DPOLI.DAT). The order in which machines are checked, however, is maintained on a lowest total processing time first priority basis. This is done so that an equitable distribution of processing will be assured for each machine of each operation. If no restrictions exist for assigning a task to a machine, then the task is scheduled on the machine by calling subroutine FILEM1.

The schedules for each machine of each operation are stored in a linked list array. Tasks are scheduled on each machine using their completion and start times. Specifically, these times are compared with the completion time of the task at the head of the current schedule. If these times are greater than or equal to this completion time, then the new task is scheduled ahead of the old task. However, if a conflict exists between the new task and the old task where the start time for the new task is less than the completion time of the old, then the new task is still scheduled ahead of the old task but the old task must be pushed back (in time) to make room for the new task. This push back procedure is performed by the subroutine PUSHBK1. The only problem that results from pushing back a task is that the raw material source for this task may not be available at the new start time. To elliminate any potential stockout situations, the program codes every task scheduled so that its raw material sources can be traced through each of the three upstream operations so that (as illustrated in figure 4.9) the completion times of each

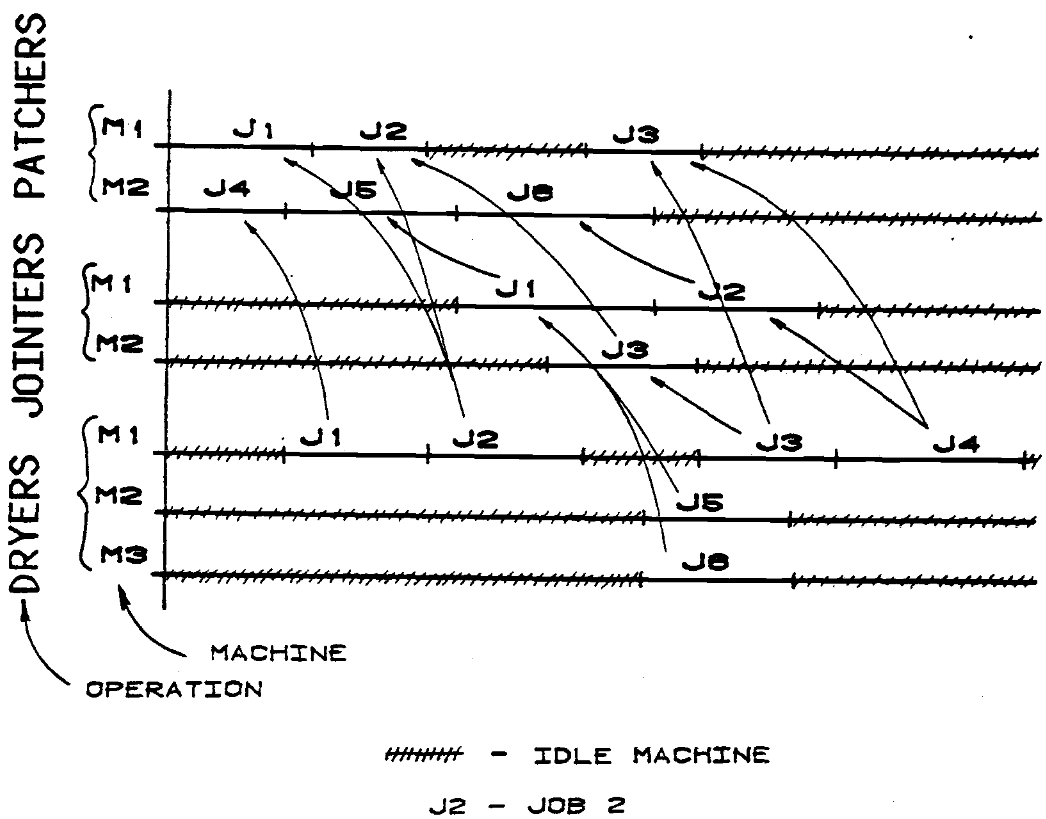


Figure 4.9 Raw material flow diagram

source can be compared with the actual times they are needed on a succeeding operation. This is done to prevent the possibility of not having an adequate supply of the appropriate raw material for the task that is being pushed back.

The trace back procedure is supported by a secondary linked storage array. Specifically, this array stores a list of raw material sources (other tasks) for each task scheduled. It is set up in subroutine FILEM2 using the row number of the primary storage array used when scheduling a task.

The routines that actually perform the check and push back of each of the raw material sources are called from subroutine PUSHBK1. These subroutines, PUSHBK2 and PUSHBK3, are only called if the stopping condition (raw material source code is equal to -1) is not met. This condition is included to prevent the program from checking tasks that are used as raw material sources for jobs on the dryer operation since this operation is considered to be a system boundary.

After a task has been successfully scheduled on the appropriate operation, the row number (or raw material source code) of the task just scheduled is sent back to the task development program. This is done to ensure that the source of the dry inventory that may be created as a result of this task will be known.

4.2.3 Phase two termination

After the entire content of the master production schedule has been filled, a termination condition is sent to the scheduling program. This termination condition trips a flag in the second program which results in a call to subroutine TALLY. This routine is called to create three task files PATTSK.DAT, JOITSK.DAT, and DRYTSK.DAT, which will be used to construct the formatted schedules for phase three.

The formatted schedules just mentioned are constructed in a third program which is run following the termination of the task development and scheduling programs. The main objective of this program is to generate a time-phased sequence of jobs for each operation so that the scheduling report can be written in phase three. As shown in figure 4.10, this program is composed of four subroutines which are driven by a main program called TSKALC. After the three schedules (PATTSK.DAT, JOITSK.DAT, and DRYTSK.DAT) have been read in program TSKALC, subroutine DAYS is called to determine the earliest start time and makespan for each operation. These statistics are in turn used by subroutine TSKFL2 to assist in time-phasing the schedules for each machine of each operation. They are also used by subroutine STIME to construct the start time file, START.DAT, that will be used in the report generation phase.

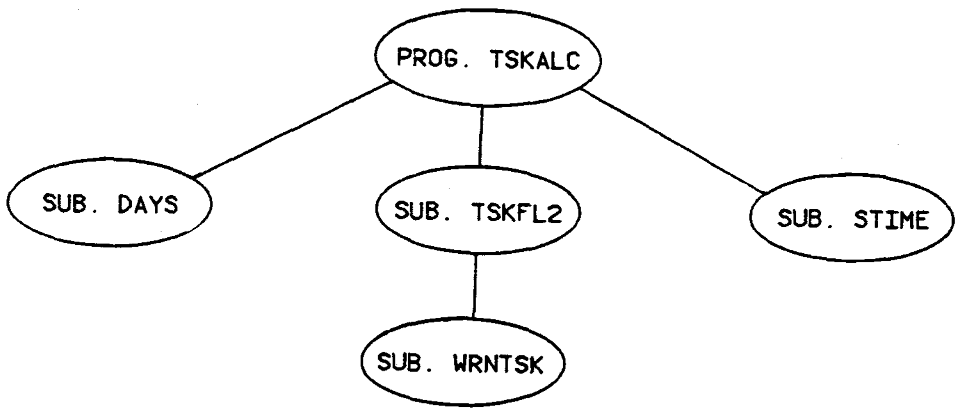


Figure 4.10 Branching tree diagram of time-phasing program

Although we have not elaborated on all of the subroutines involved in allocating tasks among the three operations, the discussion has provided a broad overview of how phase two creates and schedules tasks. Most of the other routines only provide support for those subprograms just mentioned. For more detailed information about these subroutines, refer to appendix A.

4.3 PHASE III: REPORT GENERATOR

In designing the output portion of this package, special emphasis has been given to ensure clarity and readibility of the printed results. To accomplish this goal, the body of the output is divided into four sections. Within each section, the machine schedules are listed four across a page using a GANT chart format where the y-axis represents the machines and the x-axis indicates the time. The machines are grouped this way because of the space limitations imposed by the size of the paper. If the machine capacity for an operation exceeds four, then the output is further divided into n sets of four machines and each set is printed in consecutive order (1-4 followed by 5-8, etc.).

The problem associated with developing this type of format is that for each of the four machines listed across a page, the entries of the machines schedules for each machine must line up with their corresponding starting times on the x-axis. This is partially alleviated by making the basic time interval correspond to the eight hour shift change. Therefore, the only programming concern is to ensure that the entry assigned to a given shift be in line with the other entries of the m-1 other machines. To accomplish this, statistics were maintained when the task schedules were time-phased to guarantee that the report generator includes the same number of entries for each shift among the m machines of an operation. These routines evaluate the maximum number of tasks assigned each shift over the planning horizon and store the results in one of the data files listed below.

ENTRY1.DAT - Spreader operation

ENTRY2.DAT - Patcher operation

ENTRY3.DAT - Jointer operation

ENTRY4.DAT - Dryer operation

The following discussion is a brief overview explaining how the scheduling report is generated from the sequence files, SEQ1.DAT, SEQ2.DAT, SEQ3.DAT, and SEQ4.DAT.

The report generation phase consists of nine subprograms (see figure 4.11) whose main purpose is to input and develop the set of scheduled tasks into the format mentioned above. Although program OUTPT initiates this process for each of the four operations, the central core that implements this formatting is performed by subroutine TCODE.

Specifically, the schedules for each operation are considered individually starting with the spreader operation. After the appropriate files have been opened in subroutine OPENFL, subroutine TCODE is called to read each

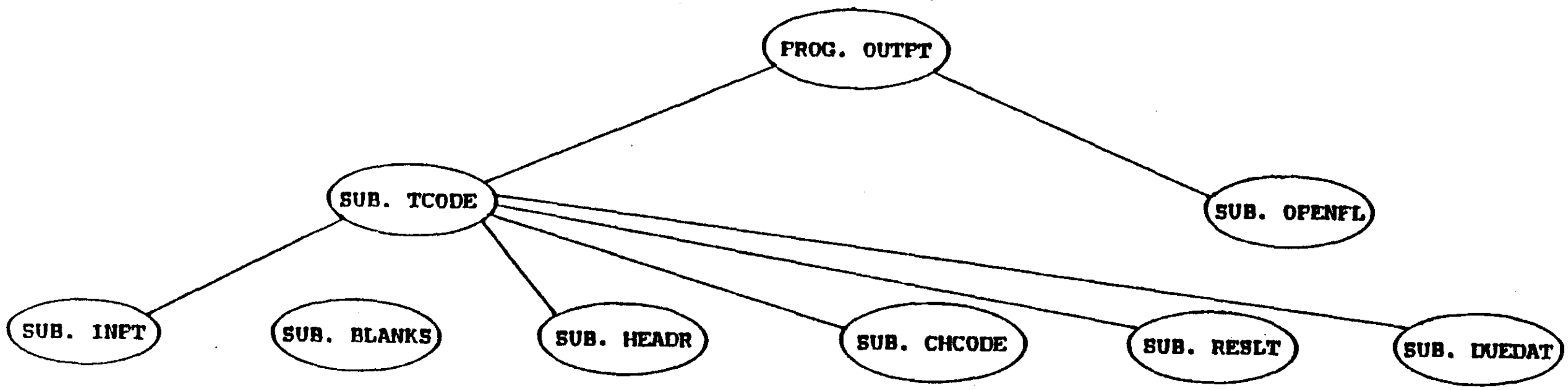


Figure 4.11 Phase three tree diagram

task from either SEQ1.DAT, SEQ2.DAT, SEQ3.DAT, or SEQ4.DAT, so that the numeric code which defines each task will be translated into a more understandable character string. The actual translation is performed by subroutine INPT. After the code is translated, subroutine TCODE then calls subroutine CHCODE to write this character string description on one of the four machines files listed below.

MACH15.DAT - Machines 1,5,9,13,or 17

MACH26.DAT - Machines 2,6,10,14,or 18

MACH37.DAT - Machines 3,7,11,15,or 19

MACH48.DAT - Machines 4,8,12,16,or 20

Outside of these primary functions, TCODE also assists in inserting the blank character strings and shift change headers in each of the machine files so that the time element is incorporated in the results. This is accomplished with subroutine BLANKS and subroutine HEADR.

After the entire task file has been translated and formatted, the content of the four formatting files is sent to the line printer by subroutine RESLT. Once this is complete, the operation counter in OUTPT is incremented and TCODE is called to start the same process again for the next operation.

CHAPTER V

RESULTS AND DISCUSSION

5.1 METHOD OF EVALUATION

The only test that could truely judge this program's effectiveness at scheduling jobs through the modeled system would be to use a benchmarking technique in which the results of a trial run are compared against a solution standard. This standard could be composed of the various aspects that are considered in evaluating a job schedule such as average inventory, maximum inventory, utilization of processing centers, and material flow. This approach would test the feasibility of each schedule as well as its merits. However, since our original objective (as stated in chapter one) only required that the program be able to construct feasible job schedules, the benchmarking method will not be used. Instead, the analysis of this package will use a more direct approach that focuses on the feasibility issue.

The feasibility question centers around whether the given schedule can actually be used without violating any of the system constraints. The first and most obvious constraint addresses whether the schedule for the three upstream operations fills the needs generated by the orders constructed on the spreader operation. Moreover, the second constraint involves the question of whether the dry inventory capacity of the system is exceeded. Another area that should be recognized when evaluating this package involves its practical use in an actual mill environment. Specifically,

1) can the package generate schedules that satisfy managements expectations in such areas as

   a. utilization of processing centers,

   b. material flow, and

   c. lead time (or time between start of dryer operation and start of spreader operation)?

and

2) how easily can the manager develop these schedules using the current package?

This second question addresses a whole new area that was not considered when developing this scheduling program. It encompasses such aspects as

1) ease of use, and

2) adaptability of the model to a dynamic mill environment.

Before any new design recommendations can be made, these practical considerations must be given immediate attention.

The evaluation of this scheduling program mainly concentrates on the feasibility issue. Two trial runs were conducted in which the results of each run were closely scrutinized with respect to the following questions:

1) Do the schedules developed for the three upstream operations truely generate the requirements needed by the spreader operation?

2) Does the dry inventory level ever exceed the system capacity at any time between the start of the dryer operation and the completion of the spreader operation?

The answers to these questions were determined by simulating an actual production run using the results (job schedules) of each trial. Specifically, the content of the master production schedule, in addition to the derived schedules for each of the three upstream operations were used as an event list (constructed in programs JTASK, DTASK, and EVNTLST). This list was then used as input to the program INVENT to evaluate the total inventory (M 3/8 surface feet basis) and the inventory level (in unit loads) of each veneer grade generated during the course of the simulation. The statistics collected included minimum and maximum inventory, and the average total inventory. These statistics were then checked to determine if any single inventory grade experienced a stock out condition. In addition, the maximum total inventory was compared with the system capacity. In either case, the schedule was considered infeasible if the computed inventory quantities violated their respective system limits.

As mentioned above, the main thrust of the analysis of this package concentrates on verifying that the schedules developed are feasible. This question only deals with whether the veneer tasks are created and scheduled correctly on the patcher, jointer, or dryer operations. What about the operation that conceives these veneer requirements? Since the spreader operation was scheduled to attempt to minimize set up time between successive tasks, the only other question that must be raised is how well does this scheduling approach compare with the methods currently employed by mill personnel?

Since the trial runs were taken from past production records, the answer to this question was determined by comparing the total set up time for the schedule composed by the package with the total set up time of the schedule developed by mill personnel. The results of this comparison in addition to the results of the feasibility question are given in the following section.

5.2 RESULTS

As indicated in table 5.1, the results for the trial one and trial two produce a maximum total inventory level of

1050.71 and 2058.61 (M 3/8).

Table 5.1 Test results

Trial	1	2
Average total Inventory M 3/8 SF	767.36	1491.54
Minimum Total Inventory M 3/8 SF	0.00	0.00
Maximum Total Inventory M 3/8 SF	1050.71	2058.61
Total Set up Time (model) MINUTES	29.50	39.00
Total Set up Time (mill) MINUTES	50.00	62.00
Lead Time DAYS	2.08	2.44
Utilization of Processing centers		
Patcher operation	.28	.54
Jointer operation	.08	.03
Dryer operation	.27	.41

respectively. These values are substantially less than the system cut off of 6000 (M 3/8). Furthermore, the inventory levels for each veneer grade also indicates that the schedule is feasible since the inventory level of every grade is either is equal to or greater than zero. These results are provided in appendix C along with the computed schedules for each operation. The set up time comparison, on the other hand, also shows a promising trend. The package in either trial sequenced jobs consistently better than the method employed by mill personnel.

5.3 DISCUSSION OF RESULTS

5.3.1 Schedule Feasibility

The results of the previous section show that in both trials the package develops feasible schedules. This conclusion is made based on the fact that neither a stock out condition exists nor the maximum total inventory level ever exceeds the system capacity during the planning period. The former conclusion can be made without reservation. That is, the scheduling program does construct viable job schedules since it was shown that the tasks that will be used to produce the desired veneer are scheduled on the appropriate operation in a way that ensures their availability on successive operations. On the other hand, the question of whether the system inventory capacity will be exceeded can not be concluded with the same conviction. In either trial, the runs were made without specifying an initial inventory. Furthermore, as seen in appendix C, the order sets which were used to schedule the spreader operation only filled this operation 40% and 42% of full capacity for each of the respective runs. Obviously, had these numbers been higher, the total inventory level may have also been higher. According to mill personnel, the average total inventory level is approximately 3 1/2 million 3/8 surface feet of veneer. This figure represents a steady state condition. The only way to determine whether inventory capacity could be exceeded would be to make a third run in which an initial inventory was given and enough orders were included to meet the capacity of the spreader operation. However, an extra run is really unnecessary since the results of the two trials (inventory results) are consistent with the expectation that management had before the runs had actually been made. Moreover, assuming that the relationship between spreader production capacity and veneer needs is linear, then the largest inventory that could have been attained if the order set required all of the available capacity (of the spreader operation) would be 2.502 million 3/8 for trial one, and 4.902 million 3/8 for trial two. These figures are still well below the system cut off of 6 million 3/8.

5.3.2 Spreader Scheduling

Aside from the feasibility aspect, the results also indicated that the use of this package for scheduling the spreader operation is promising. The results showed a 40% decrease in total set up time when using this package over the current approach employed by mill personnel.

The reasons for the large discrepancy between the two methods is most likely due to the fact that the mill scheduler is not sequencing jobs to ensure continuous material flow (or minimize total set up time). Instead, he either is scheduling the orders as they become available or is scheduling them so that each order will meet its prescribed shipment date in the finish-end. However, this does not imply that the scheduler uses only a single criterion to sequence orders. In fact, he may use several factors when scheduling.

5.3.3 Practical Useage of Program

Although we have just shown that the scheduling program can develop feasible schedules and sequence the spreaders to minimize total set up time better than the approach used by mill personnel, this still does not warrant its practical use. As stated in an earlier section, several factors must be considered before this program can actually be employed. The first criterion that must be addressed involves the quality aspect of each schedule. Do the results generated by the package satisfy the expectations of the user? For the two runs conducted, lead time was found to be 2.08 days for trial one and 2.44 days for trial two. According to management, the normal lead time period for a task is approximately one day. The difference can be attributed to the fact that tasks cannot be split among operations. It would seem that this stipulation really did not affect the results all that much because of the small difference. Utilization of processing centers, however, is another story. As shown in table 5.1, the utilization level of each operation seems to be quite low. This results from the fact that the scheduling algorithm used in phase two is not sequencing jobs to maximize machine utilization. Instead, its objective is to sequence jobs on the machines so that each sequenced task will be available at the time it is desired on a down stream operation.

The second group of practical considerations involve aspects of program design such as user friendliness, and model flexibility. At this stage, the package cannot be deemed user-friendly because it lacks a user interface. A desirable interface would be one that allows the user to run any of the three phases, and to edit or view existing input and policy files. The other aspect of program design involves model flexibility. This focuses on the models ability to encompasses changes in

1) the number of operations,
2) the number of machines used in
   each operation,

3) the panel grade knowledge base,
4) the veneer grade knowledge base (the number of thickness, length, and width categories),
5) the policy for developing a veneer grade from alternative sources, and
6) the available machine capacity of each operation.

In the present model, most of these areas cannot be modified without changing fortran code. Those, however, that can be changed include the panel grade knowledge base and the policy for developing veneer grades. To a limited extent, the available machine capacity can be modified, though, only on the spreader operation using the idle machine policy, MIDLE.DAT. A more desirable feature would give the user the ability to set any machine on any operation idle during the scheduling period.

Most of the embellishments previously cited are attainable, but before any are tried, one should recognize the limitations imposed by the computer system. The PDP 11/23 mini-computer is a multi-user system with 256k bytes of memory. However, only 64k bytes of addressable memory are allocated to a single programming task. As a result, the evolution of this scheduling program was drastically affected by the limits of the computer system. To avert any future problems, it is recommended that this package be transported to a system whose memory constraints do not prevent its expansion.

CHAPTER VI

CONCLUSIONS AND RECOMMENDATIONS

As was intended, the objective of constructing a computer program for scheduling the dry-end department of a plywood mill has been achieved. It has been shown in the preceeding chapter that in fact the package can generate feasible job schedules on each of the four operations that comprise the dry-end. The only question that does arise, though, deals mainly with the quality aspect of each schedule. It was shown that the utilization of each of the three upstream operations were quite low. However, the current scheduling heuristic does not attempt to maximize machine utilization. Instead, it concentrates solely on scheduling tasks to ensure their availability on a downstream operation. An alternative method that may prove successful at increasing utilization would be to attempt to allocate jobs to machines with large gaps of unused processing time. This method would concentrate on minimizing the makespan, thereby maximizing machine utilization. One potential drawback of this approach would be that individual veneer tasks may have to be split among machines when the tasks are actually scheduled. The current model does not support this.

The effect of minimizing makespan may have an added benefit. That is, it may also reduce the lead time period between the start of the dryer operation and the start of the spreader operation. However, this problem may be more effectively answered by addressing the stipulation that jobs must be processed in batches and batches cannot be split among operations. The current model attempts to dispel this problem by breaking large veneer orders up into suborders before scheduling so that when these smaller orders are scheduled, no single machine will be given a disproportionately large amount of total processing that may result in an unrealistic lead time. An alternative method that may prove useful to eliminate this potential lead time problem would be to violate this assumption and split batches up between successive operations. That is, schedule the start time of a task on an operation so that it will overlap a successive operation. The only pitfall of this approach would be in the evaluation of the start time. This time must be calculated so that the amount of veneer produced on a preceeding operation will be sufficient to keep the successive operation running smoothly until the remainder has been processed. Another means that may prove useful is avoiding veneer development schemes that require large amounts of an alternative veneer source to procure a rather small veneer order. An example of this situation is in developing CP veneer (of any thickness except 1/6) from a BP veneer source. Since CP veneer is produced in very small fractions (aprox. 4.5%) from BP, even with a very small CP order, the BP raw material requirement becomes exhorbitant. When carrying this BP requirement back to the dryer operation, the final veneer quantity is exagerated even more because BP is in turn a small fraction (20-36%) of another source (green inventory). The final outcome results in scheduling a task whose lead time is usually quite large. The only way this situation can be avoided, however, is to expand the list of possible veneer sources. Currently, the model only recognizes one wood species (Douglas-fir), and four veneer thicknesses (7/32, 1/6, 1/10, and 1/12). If these categories were expanded, this may assist in eliminating the potential lead time problem.

Before implementing any of these recommendations, it would be wise to invest some time in evaluating the affect that the new approach will have on computer memory. As was mentioned in earlier chapters, this package was developed on a PDP 11/23 mini-computer. The shortcomings of this system is that it can only address 64k bytes of memory for a single programming task. To avoid several hours of frustration, it is highly recommended that the programmer determine the memory requirements before emursing himself in program design.

The foundation for the scheduling program has been laid. Before this package can be considered a marketable product, the modeling limitations must be resolved. Any modifications that may be included should follow the same guidelines used in constructing this initial version. That is, the attempt here has not been to develop a scheduling program that always provides an optimal solution, but to provide a manager with a tool that gives him the ability to generate several schedules based on varying manufacturing conditions so that he may be able to make sound managerial decisions. Therefore, any subsequent changes to this program should be performed with this objective in mind.

BIBLIOGRAPHY

Baker, K. R., Introduction to Sequencing and Scheduling, John Wiley and Sons Inc., New York, 1974.

Baldwin, R. F., Plywood Manufacturing Practices, Miller Freeman Publications Inc., 1975.

Barnes, J. W., and J. J. Brennan, "Improved Algorithm for Scheduling Jobs on Identical Machines," AIIE Transactions Vol. 9 No. 1, 1977.

King, J. R., Production Planning and Control: an introduction to quantitative methods, Pergamon Press Inc., New York, 1975.

Law, A. W., and W. D. Kelton, Simulation Modeling and Analysis, McGraw-Hill Book co., New York, 1982.

McNaughton, R., "Scheduling with Deadlines and Loss Functions," Management Science Vol. 6 No. 1, Oct. 1959.

Rajaraman, M. K., "An Algorithm for Scheduling Parallel Processors," International Journal of Production Research Vol. 13 No. 5, 1975.

Root, J. G., "Scheduling with Deadlines and Loss Functions on k Parallel machines," Management Science Vol. 11 No. 3, Jan. 1965.

Rothkopf, M. H., "Scheduling Independent Tasks on Parallel Processors," Management Science Vol. 12 No. 5, 1966.

Weldwood, E. W., "Scheduling and Control of a Douglas-fir Plywood Plant Aided by Linear Programming," Forest Products Journal Vol. 12 No. 11, 1971.

Yaptenco, R. C., and A. E. Wylie, "A Quantitative Approach to Plywood Production Scheduling," Forest

Products Journal Vol. 21 No. 3, 1970.

APPENDICES

APPENDIX A

VARIABLE AND PROGRAM LISTING

APPENDIX A.1 VARIABLE DEFINITIONS AND PROGRAM LISTING FOR PHASE ONE

COL - Column number.

CPTY(i) - Machine capacity of ith operation. i=1,4.

DA - Current day.

DAT(i) - Current date. i=1,9.

IDLMS(i,j,k) - Idle machine switch used to set the ith spreader (k=1) during shift j. i=1,18, j=1,3, k=1,4.

INIT - Total number of orders assigned to a spreader of a specific grade.

MACH - Machine number.

MAXTK(i,j) - Formatting array used in phase four to ensure that when the spreader schedule is being written, the correct number of blanks are inserted in the ith shift of the jth day (if needed). i=1,3, j=1,5.

NPG - Panel grade number.

NPL - Panel length number.

NUM(i,j) - The number of orders whose attributes are grade j and length k. i=1,5, j=1,4.

NUMTSK - Total number of orders assigned to a spreader.

PN - Panel code.

PQTY - Panel order quantity.

PR(i,j,k) - Total processing time for the ith task of the jth grade and kth length category. i=1,65, j=1,5, k=1,4.

PRATE(i) - Production rate for the ith task specified in the users order file. i=1,420.

PRQTY(i) - Panel order quantity for the ith task specified in the users order file. i=1,420.

PRRGRD(i) - Panel grade priority number.

PRRLEN(i) - Panel length priority number.

FRTM(i) - Total processing time for the ith task specified in the users order file. i=1,420.

PTASK and LTASK - Six-digit panel code (3 digits each) defining a panel order.

PTLEN(i) - Total processing time required for all of the orders of the ith length category (8', 10', 12'). i=1,3.

ROW - Row number.

SCH(i) - Intermediate task sequence for a spreader. i=1,200.

SEQ(i,j,k) - Panel identity (code number) of the ith order whose attributes are grade j and length k. i=1,65, j=1,5, k=1,4.

SET(i) - Task code.

SETUP(i,j) - Set up time of the ordered task pair i,j. i=1,50, j=1,50.

TIME - Time.

TK - Task number.

TMIN - Minimum total set up time for the final task sequence.

TOTPL(i) - Total processing time available for processing all of the orders of the ith length category (8', 10', or 12'). i=1,3.

TOTPM(i) - Total processing time available on ith spreader. i=1,6.

TSEQ(i) - The final task sequence for a spreader. i=1,200.

VENGRD(i,j) - The bill of material for the ith plywood grade; this variable gives the veneer grade for the jth veneer type (face, back, center, or crossband). i=1,420, j=1,5.

VENQTY(i,j) - The bill of material for the ith plywood grade; this variable gives the amount for the jth veneer type (face, back, center, or crossband). i=1,420, j=1,5.

```
C
C    FEB. 20, 1984
C
C    THIS PROGRAM ACTS AS THE DRIVER FOR THE FIRST PHASE OF THIS THREE
C    PHASE PACKAGE.  SPECIFICALLY, THIS ROUTINE CALLS ALL OF THE SUB-
C    ROUTINES NECESSARY IN SCHEDULING THE SPREADER OPERATION.
C
C    AUTHOR:  ERIC SCHILD
C
      PROGRAM MAIN
      INTEGER VENQTY(420,5),VENGRD(420,5),TOTPM(6),PTASK,
     *IDLMS(18,3,4),LTASK,PQTY,TOTPL(3),PRODID,SHIFT,
     *MAXTK(3,5),CPTY(4),SEQ(65,5,4),NUM(5,4),PN,PTLEN(4),MACH,
     *NUMTSK,INIT,TSEQ(200),TK,DA,PRQTY(420),NTSK(3,5,6),MACHID,DAY,
     *PNUM(10),PMACH(10),PRQ(420),LFLAG,SCH(200)
      REAL PR(65,5,4),TIME,PRATE(420),PRTM(420)
      BYTE DAT(9)
      COMMON /MODEL1/ MAXTK,CPTY
      COMMON /MODEL5/ IDLMS,PTASK,LTASK,PQTY,VENQTY,VENGRD,TK,DA,
     *SEQ,NUM,PN,PR,PRQTY,NTSK,TOTPM,TOTPL,INIT,
     *MACH,NUMTSK,TIME,TSEQ,SCH,PTLEN
      COMMON /MODEL6/ PRTM,PRATE,PRQ
C
C   INITIALIZING PANEL TYPE COUNTER.
C
      DATA NUM/20*0/
C
C   SETTING UP THE MACHINE CAPACITY FOR EACH OPERATION AND
C   INITIALIZING THE PROCESSING TIME VARIABLE.
C
      DATA CPTY,PR/6,18,5,7,1300*0./
      CALL DATE(DAT)
C
C   SETTING UP THE HEADER FOR THE PANEL ORDERS.
C
      WRITE (6,5) (DAT(M),M=1,9)
5     FORMAT (T2,'CBP-LEBANON'/T2,'TASK NO. ____________',40X,
     *'SPREADER ORDER FILE',40X,9A1/)
      CALL CODINP
      CALL MACHC1
      CALL MACHTM
      CALL TSKASN
      CALL RECODE
C
C   SETTING UP THE HEADER FOR THE INITIAL INVENTORY LEVEL.
C
      WRITE (6,20) (DAT(M),M=1,9)
20    FORMAT ('1','CBP-LEBANON',40X,'CURRENT INVENTORY',40X,9A1/)
      CALL INWOOD
      STOP
      END
```

```
C
C     FEB. 20.1984
C
C     THIS SUBROUTINE IS USED TO READ THE IDLE MACHINE POLICY FILE.
C     MIDLE.
C
C     AUTHOR:  ERIC SCHILD
C
      SUBROUTINE MACHCI
      INTEGER VENQTY(420.5). VENGRD(420.5). SCH(200). IDSHF.
     *IDLMS(18.3.4). LTASK. PQTY. NTSK(3.5,6). TSEQ(200). DA. TK.
     *MAXTK(3.5). CPTY(4). SEQ(65.5.4). NUM(5.4). PN,
     *TOTPM(6). PRQTY(420). TOTPL(3). PTLEN(4). MACH. NUMTSK.
     *INIT
      REAL TIME.PR(65.5.4)
      COMMON/MODEL1/MAXTK. CPTY
      COMMON/MODEL5/IDLMS. PTASK. LTASK. PQTY. VENQTY. VENGRD. TK. DA.
     *SEQ. NUM. PN, PR. PRQTY, NTSK. TOTPM. TOTPL.
     *INIT. MACH. NUMTSK. TIME. TSEQ. SCH. PTLEN
C
      CALL ASSIGN(4.'SY:MIDLE.DAT')
      DO 25 L=1.4
C
C     READING THE IDLE MACHINE POLICY FORM MIDLE.DAT
C
      READ(4.10) ((IDLMS(J,K.L),K=1,3).J=1,CPTY(L))
10    FORMAT (60I1)
25    CONTINUE
      CALL CLOSE(4)
      RETURN
      END
```

```
C
C     FEB. 20,1984
C
C     THIS SUBPROGRAM IS USED TO TRANSLATE THE VENEER CODE AND TO WRITE
C     THE INITIAL INVENTORY LEVEL FOR THE CURRENT RUN.
C
C     AUTHOR:  ERIC SCHILD
C
      SUBROUTINE INWOOD
      INTEGER QTY,DCODE(4),LN(11)
      BYTE GRADE(11,4),THICK(11,4),WD(11,2)
      CALL ASSIGN(2,'SY:DVEN.DAT')
      K=1
C
C     READING DRY VENEER DESCRIPTION THAT WILL BE USED TO
C     TRANSLATE THE NUMERIC CODE THE USER INPUTS.
C
  5   READ(2,10,END=15) (GRADE(K,M),M=1,4),(THICK(K,M),M=1,4),
     *LN(K),(WD(K,M),M=1,2)
 10   FORMAT (T2,4A1,4A1,I2,2A1)
      K=K+1
      GO TO 5
 15   CONTINUE
      CALL CLOSE(2)
      CALL ASSIGN(2,'SY:DRY.DAT')
C
C     WRITING THE VENEER DESCRIPTION HEADER
C
      WRITE(6,20)
20    FORMAT('0','DRY INVENTORY:'/'0','GRADE  THK LENGTH WIDTH
     *QUANTITY')
C
C     READING INITIAL DRY VENEER INVENTORY
C
25    READ(2,30,END=35) (DCODE(M),M=1,4),QTY
30    FORMAT (T2,I2,3I1,I5)
C
C     WRITING DRY INVENTORY
C
      WRITE(6,27) (GRADE(DCODE(1),M),M=1,4),(THICK(DCODE(2),M),M=1,4),
     *LN(DCODE(3)),(WD(DCODE(4),M),M=1,2),QTY
27    FORMAT(T2,4A1,2X,4A1,I5,6X,2A1,I8)
      GO TO 25
35    CONTINUE
      CALL CLOSE(2)
      RETURN
      END
```

```
C
C     FEB. 20,1984
C
C     THIS SUBPROGRAM IS USED TO INPUT THE USERS ORDER FILE AND
C     TO FACILITATE THE TRANSLATION OF THE SIX DIGIT IDENTIFICATION
C     CODE SO THAT THE SCHEDULING PROCESS CAN BE INITIATED.
C
C     AUTHOR:  ERIC SCHILD
C
      SUBROUTINE CODINP
      INTEGER VENQTY(420.5),VENGRD(420.5),TOTPM(6),PTASK.
     *IDLMS(18.3.4).LTASK,PQTY.TOTPL(3),PRODID.SHIFT.
     *MAXTK(3.5).CPTY(4).SEQ(65.5.4),NUM(5.4),PN.PTLEN(4).MACH.
     *NUMTSK.INIT.TSEQ(200).TK.DA.PRQTY(420).NTSK(3,5,6).MACHID,DAY.
     *PNUM(10).PMACH(10).PRQ(420).LFLAG.NPAN.PCOD(420).LCOD(420).
     *SCH(200)
      REAL PR(65.5.4).TIME.PRATE(420).PRTM(420)
      COMMON /MODEL1/ MAXTK.CPTY
      COMMON /MODEL5/ IDLMS.PTASK.LTASK.PQTY.VENQTY.VENGRD.TK.DA.
     *SEQ.NUM.PN.PR.PRQTY.NTSK.TOTPM.TOTPL.INIT.
     *MACH.NUMTSK.TIME.TSEQ.SCH.PTLEN
      COMMON /MODEL6/ PRTM.PRATE.PRQ
C
      OPEN(UNIT=3.TYPE='OLD'.ACCESS='DIRECT'.NAME='SY:PANEL.DAT'.
     *RECORDSIZE=20.READONLY)
      CALL ASSIGN(4.'SY:PANUM.DAT')
C
C     READING IN THE 6 DIGIT CODES THAT WILL BE MATCHED WITH THE CODES
C     THE USER SPECIFIES TO DETERMINE THE INDEX NUMBER FOR ACCESSING THE
C     DESIRED PANEL GRADE IN PANEL.DAT.
C
      L=1
 5    READ (4,10.END=15) PCOD(L),LCOD(L)
10    FORMAT (4X.I4.I3)
      L=L+1
      GO TO 5
15    CONTINUE
      CALL CLOSE(4)
      CALL ASSIGN(4,'SY:TASK.DAT')
      NPAN=L-1
C
C     WRITING HEADER WHICH WILL BE USED THE DEFINE
C     THE ELEMENTS OF THE USERS ORDER FILE.
C
      WRITE (6.25)
25    FORMAT ('0'.'TASK NO. RATE          PANEL GRADE
     *'CONSTRUCTION METHOD        PRODUCTION QUANTITY'/
     *T2.'                        -------------
     *         ---------------------   ---------------------'/)
30    READ (4.35.END=75) PTASK,LTASK,PQTY
35    FORMAT (2I3.I4)
40    PN=1
C
C     CROSS REFERENCING CURRENT ORDER WITH THE 6 DIGIT CODES
C
45    IF (PN.GT.NPAN) GO TO 50
      IF (PTASK.EQ.PCOD(PN).AND.LTASK.EQ.LCOD(PN)) GO TO 50
      PN=PN+1
      GO TO 45
50    CONTINUE
```

```
C
C    CHECKING TO SEE IF ORDER HAS BEEN FOUND IN PANEL.DAT
C
        IF (PN.GT.NPAN) GO TO 55
C
C   ORDER FOUND
C
        CALL INCONS
        GO TO 30
C
C   ORDER NOT FOUND
C
55      N=1
        M=PTASK
        L=LTASK
        CALL TSKERR(N,M,L)
C
C    USER CAN RE-INPUT CORRECT ORDER
C
        READ (5,60) PTASK,LTASK,PQTY
60      FORMAT (2I3,I4)
        GO TO 40
75      CONTINUE
        CLOSE(UNIT=3)
        CALL CLOSE(4)
        RETURN
        END
```

```
C
C   FEB. 20,1984
C
C   THIS SUBPROGRAM WORKS IN CONJUNCTION WITH SUB. CODINP TO READ
C   AND CALCULATE THE STATISTICS THAT ARE NEEDED TO SCHEDULE THE SET
C   ORDERS PROVIDED BY THE USER.  THIS SUBROUTINE , SPECIFICALLY,
C   CONSTRUCTS THE BILL OF MATERIAL FOR EACH ORDER AND CALCULATES THE
C   PROCESSING TIMES THAT ARE VITAL IN SEQUENCING THE TASKS ON EACH
C   MACHINE.
C
C   AUTHOR:  ERIC SCHILD
C
      SUBROUTINE INCONS
      INTEGER VENQTY(420,5),VENGRD(420,5),TOTPM(6),PTASK,
     *IDLMS(18,3,4),LTASK,PQTY,TOTPL(3),PRODID,SHIFT,
     *MAXTK(3,5),CPTY(4),SEQ(65,5,4),NUM(5,4),PN,PTLEN(4),MACH,
     *NUMTSK,INIT,TSEQ(200),TK,DA,PRQTY(420),NTSK(3,5,6),MACHID,DAY,
     *PNUM(10),PMACH(10),PRQ(420),LFLAG,NP,LN,PLYS,CONST,LEN,
     *SCH(200)
      REAL PR(65,5,4),TIME,PRATE(420),PRTM(420)
      BYTE GRADE(8),THK(5),EXT(3),TYPE(3)
      COMMON /MODEL1/ MAXTK,CPTY
      COMMON /MODEL5/ IDLMS,PTASK,LTASK,PQTY,VENQTY,VENGRD,TK,DA,
     *SEQ,NUM,PN,PR,PRQTY,NTSK,TOTPM,TOTPL,INIT,
     *MACH,NUMTSK,TIME,TSEQ,SCH,PTLEN
      COMMON /MODEL6/ PRTM,PRATE,PRQ
C
      DO 35 I=1,5
C
C   DETERMINING THE PANEL GRADE FOR THE CURRENT ORDER.
C
      IF (PTASK.GT.(I+1)*100) GO TO 35
      DO 25 J=1,4
C
C   DETERMINING THE PANEL LENGTH FOR THE CURRENT ORDER.
C
      IF (LTASK.GT.(J+1)*100) GO TO 25
C
C   INCREMENTING PANEL TYPE COUNTER.
C
      NUM(I,J)=NUM(I,J)+1
C
C   RETAINING INDEX THAT WILL BE USED THE IDENTIFY EACH USER
C   DEFINED TASK.
C
      SEQ(NUM(I,J),I,J)=PN
      READ (3'PN) NP,LN,PLYS,PRATE(PN),CONST,LEN,(TYPE(M),M=1,3),
     *(GRADE(M),M=1,8),(EXT(M),M=1,3),(THK(M),M=1,5),
     *(VENQTY(PN,M),M=1,5),(VENGRD(PN,M),M=1,5)
C
C   PROCESSING TIME CALCULATION
C
      PR(NUM(I,J),I,J)=FLOAT(PQTY)/PRATE(PN)
      PRQTY(PN)=PQTY
      WRITE (6,15) NP,LN,PRATE(PN),LEN,(TYPE(M),M=1,3),
     *(GRADE(M),M=1,8),(EXT(M),M=1,3),(THK(M),M=1,5),
     *(VENQTY(PN,M),M=1,5),(VENGRD(PN,M),M=1,5),PQTY
 15   FORMAT (I2,2I3,2X,F5.2,2X,I3,1X,3A1,1X,8A1,1X,3A1,1X,5A1,1X,
     *I3,4I2,5I4,10X,I5)
      RETURN
```

```
25    CONTINUE
35    CONTINUE
      RETURN
      END
```

```
C
C    FEB. 20.1984
C
C    THIS SUBPROGRAM WRITES THE ERROR MESSAGES FOR THE TWO ERROR
C    CHECKS THAT ARE MADE IN PHASE ONE.
C
C    AUTHOR:  ERIC SCHILD
C
      SUBROUTINE TSKERR(K,L,M)
      INTEGER K,L,M
      IF(K.GT.10)GO TO 15
      WRITE (5,5) K,L,M
 5    FORMAT (T2,'ERROR(',I1,')  BAD TASK CODE-',2I3)
      WRITE (5,10)
 10   FORMAT (T2,'RE-ENTER CODE'/T2,'=',$)
      RETURN
 15   IF (K.GT.20)RETURN
      N=K-10
      IF (N.EQ.4) J=12
      IF (N.EQ.3) J=10
      IF (N.EQ.1) J=8
      WRITE (5,20) K,J,L
 20   FORMAT (T2,'ERROR(',I2,')',I3,' FT. PRODUCTION EXCEEDS'/T2,
     *'MACHINE CAPACITY BY',I4,' MIN.')
      RETURN
      END
```

```
C
C     FEB. 20,1984
C
C     THIS SUBPROGRAM USES THE IDLE MACHINE POLICY READ IN FROM SUB.
C     MACHC1 TO COMPUTE THE AVAILABLE PROCESSING TIME FOR EACH MACHINE
C     OF THE SPREADER OPERATION.
C
C     AUTHOR:  ERIC SCHILD
C
      SUBROUTINE MACHTM
      INTEGER VENQTY(420,5),VENGRD(420,5),TOTPM(6),PTASK,
     *IDLMS(18,3,4),LTASK,PQTY,TOTPL(3),PRODID,SHIFT,
     *MAXTK(3,5),CPTY(4),SEQ(65,5,4),NUM(5,4),PN,PILEN(4),MACH,
     *NUMTSK,INIT,TSEQ(200),TK,DA,PRQTY(420),NTSK(3,5,6),MACHID,DAY,
     *PNUM(10),PMACH(10),PRQ(420),LFLAG,PRESTP(6),MASPAN,MA,SH,
     *RSHIFT,SCH(200)
      REAL PR(65,5,4),TIME,PRATE(420),PRTM(420)
      COMMON /MODEL1/ MAXTK,CPTY
      COMMON /MODEL5/ IDLMS,PTASK,LTASK,PQTY,VENQTY,VENGRD,TK,DA,
     *SEQ,NUM,PN,PR,PRQTY,NTSK,TOTPM,TOTPL,INIT,
     *MACH,NUMTSK,TIME,TSEQ,SCH,PILEN
      COMMON /MODEL6/ PRTM,PRATE,PRQ
C
      DATA PRESTP/1,3,2,1,2,3/,TOTPL/3*0/,MASPAN/5/
C
      DO 25 MA=1,6
C
C     INITIALIZING IDLE SHIFT COUNTER
C
      RSHIFT=0
C
C     COUNTING THE NUMBER OF IDLE SHIFTS FOR EACH MACHINE.
C
      DO 20 SH=1,3
      IF (IDLMS(MA,SH,1).EQ.0) GO TO 20
      RSHIFT=RSHIFT+1
 20   CONTINUE
C
C     DETERMINING TOTAL PROCESSING TIME FOR EACH MACHINE.
C
      TOTPM(MA)=RSHIFT*480*MASPAN
C
C     DETERMINING TOTAL PROCESSING TIME FOR EACH MACHINE CATEGORY
C                   MACHINE 2, & 6- 12 FOOT ONLY
C               MACHINE 1, & 4- 8 FOOT ONLY
C               MACHINE 3, & 5- 10 FOOT ONLY
C
      TOTPL(PRESTP(MA))=TOTPL(PRESTP(MA))+TOTPM(MA)
 25   CONTINUE
      CALL TASKTM
      CALL ERRDET
      RETURN
      END
```

```
C
C    FEB. 20,1984
C
C   LIKE SUB. MACHTM, THIS SUBPROGRAM COMPUTES THE TOTAL PROCESSING
C   NEEDS FOR THE SET OF ORDER SPECIFIED BY THE USER.  TO REMAIN CON-
C   SISTENT WITH SUB. MACHTM, THE TOTAL PROCESSING TIME IS PARTITIONED
C   INTO FOUR CATEGORIES (BASED ON PANEL LENGTH).
C
C    AUTHOR:  ERIC SCHILD
C
      SUBROUTINE TASKTM
      INTEGER VENQTY(420,5),VENGRD(420,5),TOTPM(6),PTASK,
     *IDLMS(18,3,4),LTASK,PQTY,TOTPL(3),PRODID,SHIFT,
     *MAXTK(3,5),CPTY(4),SEQ(65,5,4),NUM(5,4),PN,PTLEN(4),MACH,
     *NUMTSK,INIT,TSEQ(200),TK,DA,PRQTY(420),NTSK(3,5,6),MACHID,DAY,
     *PNUM(10),PMACH(10),PRQ(420),LFLAG,PL,PG,ITEM,SCH(200)
      REAL PR(65,5,4),TIME,PRATE(420),PRTM(420)
      COMMON /MODEL1/ MAXTK,CPTY
      COMMON /MODEL5/ IDLMS,PTASK,LTASK,PQTY,VENQTY,VENGRD,TK,DA,
     *SEQ,NUM,PN,PR,PRQTY,NTSK,TOTPM,TOTPL,INIT,
     *MACH,NUMTSK,TIME,TSEQ,SCH,PTLEN
      COMMON /MODEL6/ PRTM,PRATE,PRQ
C
C    INITIALIZING TOTAL PROCESSING COUNTER
C
      DATA PTLEN/4*0/
C
      DO 30 PL=1,4
C
      DO 25 PG=1,5
C
C     DETERMINING THE TOTAL PROCESSING TIME FOR EACH LENGTH CATEGORY
C
      DO 20 ITEM=1,NUM(PG,PL)
      PTLEN(PL)=PTLEN(PL)+PR(ITEM,PG,PL)
 20   CONTINUE
C
 25   CONTINUE
C
 30   CONTINUE
      RETURN
      END
```

```
C
C    FEB. 20,1984
C
C    THIS SUBPROGRAM EVALUATES THE RESULTS OF SUB. MACHTM AND SUB.
C    TASKTM TO DETERMINE WHETHER THE USER HAS OVERLOADED THE SYSTEM
C    WITH ORDERS.  IF HE HAS, THEN AN ERROR MESSAGE IS WRITTEN BY
C    CALLING SUB. TSKERR.
C
C    AUTHOR:  ERIC SCHILD
C
      SUBROUTINE ERRDET
      INTEGER VENQTY(420,5),VENGRD(420,5),TOTPM(6),PTASK,
     *IDLMS(18,3,4),LTASK,PQTY,TOTPL(3),PRODID,SHIFT,
     *MAXTK(3,5),CPTY(4),SEQ(65,5,4),NUM(5,4),PN,PTLEN(4),MACH,
     *NUMTSK,INIT,TSEQ(200),TK,DA,PRQTY(420),NTSK(3,5,6),MACHID,DAY,
     *PNUM(10),PMACH(10),PRQ(420),LFLAG,P8,P109,P12,SCH(200)
      REAL PR(65,5,4),TIME,PRATE(420),PRTM(420)
      COMMON /MODEL1/ MAXTK,CPTY
      COMMON /MODEL5/ IDLMS,PTASK,LTASK,PQTY,VENQTY,VENGRD,TK,DA,
     *SEQ,NUM,PN,PR,PRQTY,NTSK,TOTPM,TOTPL,INIT,
     *MACH,NUMTSK,TIME,TSEQ,SCH,PTLEN
      COMMON /MODEL6/ PRTM,PRATE,PRQ
C
C    COMPARING TOTAL PROCESSING TIME FOR 12' PANELS WITH THE AVAILABLE
C    PROCESSING TIME ON MACHINES 2 AND 6.
C
      IF (PTLEN(4).GT.TOTPL(3)) GO TO 5
      P12=TOTPL(3)-PTLEN(4)
      P109=TOTPL(2)+P12
C
C    COMPARING TOTAL PROCESSING TIME FOR 10' AND 9' PANELS WITH THE
C    REMAINING TIME ON MACHINE 2 AND 6, AND THE AVAILABLE TIME
C    ON MACHINE 3 AND 5.
C
      IF (PTLEN(2)+PTLEN(3).GT.P109) GO TO 10
      P8=P109-(PTLEN(2)+PTLEN(3))+TOTPL(1)
C
C    COMPARING TOTAL PROCESSING TIME FOR 8' PANELS WITH THE AVAILABLE
C    PROCESSING TIME ON THE REMAINING MACHINES.
C
      IF (PTLEN(1).GT.P8) GO TO 15
      RETURN
C
C    ORDER FILE EXCEEDS SYSTEM CAPACITY
C
 5    N=14
      M=PTLEN(4)-TOTPL(3)
      CALL TSKERR(N,M)
      RETURN
C
 10   N=13
      M=PTLEN(2)+PTLEN(3)-P109
      CALL TSKERR(N,M)
      RETURN
C
 15   N=11
      M=PTLEN(1)-P8
      CALL TSKERR(N,M)
      RETURN
      END
```

```
C
C     FEB. 20,1984
C
C     THIS SUBPROGRAM USES THE PANEL TYPE COUNTER, THE GLUE SPREADER
C     ALLOCATION POLICY (GLPOLI.DAT), AND THE PROCESSING TIMES COMPUTED
C     IN SUB. INCONS TO ALLOCATE THE ORDERS AMONG THE SIX MACHINES OF THE
C     SPREADER OPERATION.
C
C     AUTHOR:  ERIC SCHILD
C
      SUBROUTINE TSKASN
      INTEGER VENQTY(420,5),VENGRD(420,5),TOTPM(6),PTASK,TMIN,
     *IDLMS(18,3,4),LTASK,PQTY,TOTPL(3),PRODID,SHIFT,RMIN,
     *MAXTK(3,5),CPTY(4),SEQ(65,5,4),NUM(5,4),PN,PTLEN(4),MACH,
     *NUMTSK,INIT,TSEQ(200),TK,DA,PRQTY(420),NTSK(3,5,6),MACHID,DAY,
     *PNUM(10),PMACH(10),PRQ(420),LFLAG,PRRGRD(5),PRRLEN(4),
     *PG,PL,NPG,NPL,GRAD,COUNT,SCH(200),SETUP(50,50),ROW,COL,SET(420),
     *INDEX
      REAL PR(65,5,4),TIME,PRATE(420),TOTP,PRTM(420)
      LOGICAL FLAG
      COMMON /MODEL1/ MAXTK,CPTY
      COMMON /MODEL5/ IDLMS,PTASK,LTASK,PQTY,VENQTY,VENGRD,TK,DA,
     *SEQ,NUM,PN,PR,PRQTY,NTSK,TOTPM,TOTPL,INIT,
     *MACH,NUMTSK,TIME,TSEQ,SCH,PTLEN
      COMMON /MODEL6/ PRTM,PRATE,PRQ
      COMMON /MODEL7/ SETUP,ROW,COL,SET,INDEX,TMIN
C
      OPEN(UNIT=2,TYPE='OLD',ACCESS='DIRECT',NAME='SY:GLPOLI.DAT',
     *RECORDSIZE=10,READONLY)
      CALL ASSIGN(4,'SY:SEQ1.DAT')
      CALL ASSIGN(1,'SY:PAT.DAT')
C
C     INITIALIZING THE FORMATTING COUNTER THAT WILL BE USED IN
C     PHASE FOUR TO FORMAT THE SCHEDULED LIST.
C
      DATA MAXTK/15*0/
      RMIN=0
      CALL SETR
C
      DO 90 MACH=1,6
C
C     SETTING UP TASK ALLOCATION POLICY
C
      READ(2'MACH) (PRRGRD(M),M=1,5),(PRRLEN(M),M=1,4)
C
C      AVAILABLE PROCESSING TIME ON THE CURRENT MACHINE.
C
      TOTP=FLOAT(TOTPM(MACH))
C
C     INITIALIZING TASK ASSIGNMENT COUNTER
C
      NUMTSK=0
      FLAG=.FALSE.
C
      IF (TOTP.GT.0) GO TO 8
      CALL TSKFL1
      GO TO 90
C
C     SETTING UP THE LOOP THAT WILL ASSIST IN CHECKING THE PANEL
C     GRADES THAT CAN BE ASSIGNED TO THE CURRENT MACHINE.
```

```
C
 8      DO 60 PG=1,5
C
C     SETTING UP THE LOOP THAT WILL ASSIST IN CHECKING THE PANEL
C     LENGTHS THAT CAN BE ASSIGNED TO THE CURRENT MACHINE.
C
        DO 55 PL=1,4
        NPG=PRRGRD(PG)
        NPL=PRRLEN(PL)
C
C     CHECKING TO SEE IF ANY ORDERS ARE AVAILABLE UNDER THE CURRENT GRADE-
C     LENGTH CATEGORY
C
        IF (NUM(NPG,NPL).EQ.0) GO TO 55
C
C     CHECKING TO SEE IF THIS GRADE-LENGTH CATEGORY IS THE FIRST TO BE
C     ALLOCATED TO THE CURRENT MACHINE.
C
        IF (NUMTSK.GT.0) GO TO 30
        GRAD=NPG
C
C     CHECKING TO DETERMINE WHEN CURRENT ALLOCATED GRADE CHANGES.  WHEN IT
C     DOES THE VALUE OF NUMTSK IS RETAINED FOR THE SEQUENCING ROUTINE.
C
 30     IF (NPG.EQ.GRAD) GO TO 35
        IF(FLAG)GO TO 35
        INIT=NUMTSK
        FLAG=.TRUE.
  35    NM=NUM(NPG,NPL)
C
C     ENUMERATING ALL OF THE ORDERS IN THE CURRENT GRADE-LENGTH CATEGORY.
C
        DO 50 COUNT=1,NM
C
C     CHECKING PROCESSING TIME OF CURRENT ORDER.
C
        IF (PR(COUNT,NPG,NPL).EQ.0) GO TO 50
        NUMTSK=NUMTSK+1
C
C     SETTING ASSIGNMENT FLAG
C
        LFLAG=0
C
C     SETTING UP ASSIGNMENT STATISTICS FOR CURRENT ORDER.
C
        SCH(NUMTSK)=SEQ(COUNT,NPG,NPL)
        PRTM(SCH(NUMTSK))=PR(COUNT,NPG,NPL)
        PRQ(SCH(NUMTSK))=PRQTY(SCH(NUMTSK))
        PR(COUNT,NPG,NPL)=AMAX1(0.,PRTM(SCH(NUMTSK))-TOTP)
        PRQTY(SCH(NUMTSK))=MAX1(0.,FLOAT(PRQ(SCH(NUMTSK)))-TOTP*PRATE
       *(SCH(NUMTSK)))
C
C     CHECKING TO SEE IF THE PROCESSING TIME REQUIRED FOR THE CURRENT ORDER
C     IS GREATER THAN THE TIME AVAILABLE ON THE CURRENT MACHINE.
C
        IF (PRTM(SCH(NUMTSK)).GT.TOTP) GO TO 61
        TOTP=TOTP-PRTM(SCH(NUMTSK))
        LFLAG=1
C
 50     CONTINUE
```

```
C
 55   CONTINUE
C
 60   CONTINUE
C
C    CHECKING TO SEE IF THE ORDER FILE HAS BEEN EXHAUSTED BEFORE
C    FILLING THE CURRENT MACHINE.
C
 61   IF (LFLAG.EQ.1) GO TO 62
      PRTM(SCH(NUMTSK))=TOTP
      PRQ(SCH(NUMTSK))=IFIX(TOTP*PRATE(SCH(NUMTSK)))
 62   CALL TSKSEQ
      RMIN=RMIN+TMIN
      CALL TSKFL1
      CALL MAXTSK
 90   CONTINUE
      CLOSE(UNIT=2)
      CALL CLOSE(4)
      CALL CLOSE(1)
      CALL WRNTSK
      CALL ASSIGN(1,'SY:SETUP.DAT')
      AMIN=FLOAT(RMIN)/60.
      WRITE(1,95) AMIN
95    FORMAT(T2,F7.2)
      CALL CLOSE(1)
      RETURN
      END
```

```
C
C     FEB. 20,1984
C
C     THIS SUBPROGRAM IS USED TO READ IN THE USER WRITTEN FILE.
C     STIME.DAT. WHICH DESIGNATES THOSE TASK PAIRS THAT WILL HAVE
C     EXCESSIVELY HIGH SET-UP TIMES.  THIS FILE WILL BE USED IN
C     CONJUNCTION WITH SUB. MATRX1 AND SUB. MATRX2 TO CONSTRUCT THE
C     SET-UP TIME MATRIX FOR SEQUENCING THE TASK LIST ON THE CURRENT
C     MACHINE.
C
C     AUTHOR:  ERIC SCHILD
C
      SUBROUTINE SETR
      INTEGER SETUP(50,50),ROW,COL,SET(420),INDEX,TMIN
      COMMON /MODEL7/ SETUP,ROW,COL,SET,INDEX,TMIN
      CALL ASSIGN(3,'SY:STIME.DAT')
C
      INDEX=1
5     READ(3,10,END=15) SET(INDEX)
10    FORMAT(T2,I3)
      INDEX=INDEX+1
      GO TO 5
15    CONTINUE
      INDEX=INDEX-1
      CALL CLOSE(3)
      RETURN
      END
```

```
C
C     FEB. 20,1984
C
C     THIS SUBPROGRAM EVALUATES THE FORMATTING VARIABLES (DETERMINED IN
C     SUB. TSKFIL) TO FIND THE LARGEST NUMBER OF TASKS ASSIGNED TO A
C     SHIFT AMONG THE SIX MACHINES.
C
C     AUTHOR:  ERIC SCHILD
C
        SUBROUTINE MAXTSK
        INTEGER VENQTY(420.5),VENGRD(420.5),TOTPM(6),PTASK.
       *IDLMS(18,3,4),LTASK.PQTY.TOTPL(3),PRODID.SHIFT,
       *MAXTK(3.5).CPTY(4),SEQ(65.5.4).NUM(5.4),PN,PTLEN(4).MACH,
       *NUMTSK.INIT,TSEQ(200).TK.DA.PRQTY(420).NTSK(3,5,6).MACHID,DAY.
       *PNUM(10),PMACH(10).PRQ(420).LFLAG.SH.SCH(200)
        REAL PR(65.5.4).TIME.PRTM(420).PRATE(420)
        COMMON /MODEL1/ MAXTK.CPTY
        COMMON /MODEL5/ IDLMS.PTASK.LTASK,PQTY,VENQTY.VENGRD.TK.DA.
       *SEQ.NUM,PN.PR.PRQTY,NTSK,TOTPM.TOTPL.INIT,
       *MACH.NUMTSK.TIME,TSEQ.SCH.PTLEN
        COMMON /MODEL6/ PRTM.PRATE.PRQ
C
        DO 80 DA=1,5
C
        DO 75 SH=1.3
C
C     COMPARING THE CURRENT NUMBER OF TASKS ASSIGNED TO A MACHINE ON A
C     GIVEN SHIFT WITH THE MAXIMUM NUMBER ALREADY FOUND.
C
        IF (NTSK(SH.DA.MACH).LT.MAXTK(SH.DA)) GO TO 75
        MAXTK(SH,DA)=NTSK(SH,DA,MACH)
 75     CONTINUE
C
 80     CONTINUE
        RETURN
        END
```

```
C
C    FEB. 20.1984
C
C    THIS SUBPROGRAM DEVELOPS THE FORMATTING FILE. ENTRY1.DAT. THAT WILL
C    BE USED IN PHASE FOUR TO ASSIST IN THE CONSTRUCTION OF THE SCHEDUL-
C    ING REPORT FOR THE SPREADER OPERATION.
C
C    AUTHOR: ERIC SCHILD
C
      SUBROUTINE WRNTSK
      INTEGER VENQTY(420.5).VENGRD(420.5).TOTPM(6).PTASK,
     *IDLMS(18,3,4).LTASK.PQTY.TOTPL(3).PRODID,SHIFT.
     *MAXTK(3.5),CPTY(4),SEQ(65.5.4).NUM(5.4).PN.PTLEN(4).MACH.
     *NUMTSK.INIT.TSEQ(200).TK.DA.PRQTY(420).NTSK(3.5.6).MACHID.DAY.
     *PNUM(10).SH.PMACH(10).PRQ(420).LFLAG.SCH(200)
      REAL PR(65.5.4).TIME.PRTM(420).PRATE(420)
      COMMON /MODEL1/ MAXTK.CPTY
      COMMON /MODEL5/ IDLMS.PTASK.LTASK,PQTY.VENQTY.VENGRD.TK.DA.
     *SEQ.NUM.PN.PR.PRQTY.NTSK,TOTPM.TOTPL.INIT.
     *MACH.NUMTSK.TIME.TSEQ.SCH.PTLEN
      COMMON /MODEL6/ PRTM.PRATE.PRQ
C
      CALL ASSIGN(1,'SY:ENTRY1.DAT')
      DO 25 DA=1.5
      WRITE(1,10) (MAXTK(SH,DA),SH=1.3)
10    FORMAT(T2.3I2)
25    CONTINUE
      CALL CLOSE(1)
      RETURN
      END
```

```
C
C     FEB. 20,1984
C
C     THIS SUBPROGRAM IS USED TO TIME PHASE THE NEWLY SEQUENCED ORDERS
C     OVER THE 5 DAY PLANNING HORIZON AND TO WRITE THIS TIME PHASED
C     SEQUENCE ON SEQ4.DAT.  IT IS ALSO USED TO ASSIST IN CREATING
C     PAT.DAT BY CALLING SUB. PATFIL AFTER EACH TASK HAS BEEN TIME
C     PHASED.
C
C     AUTHOR:  ERIC SCHILD
C
      SUBROUTINE TSKFL1
      INTEGER VENQTY(420,5),VENGRD(420,5),TOTPM(6),PTASK,
     *IDLMS(18,3,4),LTASK,TOTPL(3),PRODID,SHIFT,
     *MAXTK(3,5),CPTY(4),SEQ(65,5,4),NUM(5,4),PN,PTLEN(4),MACH,
     *NUMTSK,INIT,TSEQ(200),TK,DA,PRQTY(420),NTSK(3,5,6),MACHID,DAY,
     *PNUM(10),PMACH(10),PRQ(420),LFLAG,SH,PRID,PQTY
      REAL PR(65,5,4),PRATE(420),NXT(3),TIME,PRTM(420)
      COMMON /MODEL1/ MAXTK,CPTY
      COMMON /MODEL5/ IDLMS,PTASK,LTASK,PQTY,VENQTY,VENGRD,TK,DA,
     *SEQ,NUM,PN,PR,PRQTY,NTSK,TOTPM,TOTPL,INIT,
     *MACH,NUMTSK,TIME,TSEQ,SCH,PTLEN
      COMMON /MODEL6/ PRTM,PRATE,PRQ
C
C    INITIALIZING TASK COUNTER AND SETTING SHIFT CHANGE VARIABLES
C
      DATA NTSK/90*0/
      TK=1
      PFLAG=0
      DATA NXT/480.,960.,1440./
C
      DO 40 DA=1,5
      TIME=0.
      DO 35 SH=1,3
C
C    CHECKING IDLE MACHINE VARIABLE TO DETERMINE WHETHER THE MACHINE WILL
C    BE SET IDLE ON THE CURRENT SHIFT.
C
      IF (IDLMS(MACH,SH,1).GT.0) GO TO 10
      PRID=-1
      PQTY=0
C
C    WRITING TASK ON SEQ4.DAT
C
      WRITE (4,45) PRID,PQTY,MACH,SH
      TIME=NXT(SH)
      PRID=TSEQ(TK)
      GO TO 35
C
 10   IF (PFLAG.GT.0) GO TO 15
C
C    CHECKING TO SEE IF ALL THE ALLOCATED ORDERS HAVE BEEN TIME PHASED
C
      IF (TK.GT.NUMTSK) GO TO 25
      PRID=TSEQ(TK)
 15   PFLAG=0
C
C    CHECKING WHETHER THE PROCESSING TIME FOR THE CURRENT ORDER IS ENOUGH
C    TO FILL CURRENT SHIFT.
C
```

```
      IF (PRTM(PRID)+TIME.LT.NXT(SH)) GO TO 20
      PQTY=IFIX((NXT(SH)-TIME)*PRATE(PRID))
      PRQ(PRID)=PRQ(PRID)-PQTY
      PRTM(PRID)=PRTM(PRID)-(NXT(SH)-TIME)
      CALL PATFIL
      TIME=NXT(SH)
      NTSK(SH,DA,MACH)=NTSK(SH,DA,MACH)+1
      PFLAG=1
C
C    WRITING THE TASK ON SEQ4.DAT
C
      WRITE (4,45) PRID,PQTY,MACH,SH
      GO TO 35
C
 20   PRID=TSEQ(TK)
      PQTY=PRQ(PRID)
      CALL PATFIL
      TIME=TIME+PRTM(PRID)
      NTSK(SH,DA,MACH)=NTSK(SH,DA,MACH)+1
C
C    WRITING THE TASK ON SEQ4.DAT
C
      WRITE (4,45) PRID,PQTY,MACH,SH
      TK=TK+1
      GO TO 10
C
C    CREATING IDLE SHIFTS IF ALLOCATED ORDER LIST IS EXHAUSTED BEFORE
C    THE SCHEDULING PERIOD (5 DAYS) IS COMPLETE.
C
 25   IF (NUMTSK.GT.0) GO TO 28
      PRID=-1
      PQTY=0
      WRITE (4,45) PRID,PQTY,MACH,SH
 28   DO 30 I=1,3
      IDLMS(MACH,I,1)=0
 30   CONTINUE
 35   CONTINUE
 40   CONTINUE
 45   FORMAT (I4,I10,I3,I4)
      RETURN
      END
```

```
C
C     FEB. 20,1984
C
C     THIS SUBPROGRAM USES THE BILL OF MATERIAL AND THE STARTING TIMES
C     FOR EACH ORDER (DETERMINED IN SUB. TSKFIL) TO CONSTRUCT THE
C     MASTER PRODUCTION SCHEDULE. PAT.DAT. FOR PHASE TWO.
C
C     AUTHOR:  ERIC SCHILD
C
        SUBROUTINE PATFIL
        INTEGER VENQTY(420,5).VENGRD(420.5).TOTPM(6).PTASK.
      *IDLMS(18.3.4).LTASK,PQTY,TOTPL(3),PRODID,SHIFT,
      *MAXTK(3.5),CPTY(4),SEQ(65,5,4),NUM(5.4).PN,PTLEN(4).MACH.
      *NUMTSK.INIT.TSEQ(200),TK,DA,PRQTY(420).NTSK(3,5.6).MACHID.DAY.
      *PNUM(10).PMACH(10),PRQ(420).LFLAG.TIM.SCH(200).VENQ.
      *TYPE.ROUND,LEN
        REAL PR(65.5,4).TIME.PRTM(420).PRATE(420)
        COMMON /MODEL1/ MAXTK.CPTY
        COMMON /MODEL5/ IDLMS,PTASK.LTASK.PQTY.VENQTY.VENGRD.TK.DA.
      *SEQ.NUM,PN.PR.PRQTY.NTSK.TOTPM.TOTPL.INIT.
      *MACH,NUMTSK.TIME,TSEQ.SCH.PTLEN
        COMMON /MODEL6/ PRTM.PRATE.PRQ
C
C     RECOMPUTING THE STARTING TIME FOR EACH ORDER AND
C     DETERMINING THE LENGTH OF THE ORDER.
C
        TIM=IFIX((DA-1)*1440.+TIME)
        ROUND=INT(FLOAT(VENGRD(TSEQ(TK),1))/10.)*10
        LEN=VENGRD(TSEQ(TK).1)-ROUND
C
C     DEVELOPING VENEER REQUIREMENTS (FACE. BACK. CNTR. AND XBND) FOR
C     EACH ORDER.
C
        DO 10 K=1.5
C
C     DESIGNATING WHETHER THE ORDER IS A FACE. BACK. CNTR.
C     OR XBND.
C
        TYPE=K
C
C     COMPUTING THE NEEDED QUANTITY FOR THIS VENEER CATEGORY.
C
        VENQ=VENQTY(TSEQ(TK).K)*PQTY
        IF (VENQ.EQ.0) GO TO 10
C
C     IF VENEER GRADE INVOLVES A FULL WIDTH SHEET, CONTINUE
C
        IF(VENGRD(TSEQ(TK).K).GE.100)GO TO 7
        TYPE=4
C
C     ADDING THE PANEL LENGTH VARIABLE TO THE XBND VENEER CODE.
C
        NVAL=VENGRD(TSEQ(TK).K)*10+LEN
        WRITE (1.15) NVAL,VENQ.TIM.TYPE
        GO TO 10
7       IF(K.GT.3)TYPE=3
        WRITE (1.15) VENGRD(TSEQ(TK).K).VENQ.TIM.TYPE
 10     CONTINUE
 15     FORMAT (I2.I3.I5.I6,I2)
        RETURN
```

END

```
C
C    FEB. 20,1984
C
C    THIS SUBPROGRAM SEQUENCES THE NEWLY ALLOCATED ORDERS ON THE CURRENT
C    MACHINE, INDICATED BY VARIABLE MACH, ACCORDING TO THE DEPENDENT
C    SET UP TIME HEURISTIC.  IT USES THE MODIFIED ALL ORIGINS APPROACH
C    IN WHICH ONLY THE FIRST INIT ORDERS WILL BE USED AS ORIGINS.
C
C    AUTHOR:  ERIC SCHILD
C
      SUBROUTINE TSKSEQ
      INTEGER VENQTY(420,5), VENGRD(420,5), PQTY, TOTPM(6), SEQ(65,5,4),
     *NUM(5,4), PN, IDLMS(18,3,4), PTASK, LTASK, NTSK(3,5,6), MACH, DA,
     *PRQTY(420), TOTPL(3), TK, SCH(200), P,
     *PTLEN(4), NUMTSK, INIT, TSEQ(200), SQ, TSCH(200), TMIN,
     *TOTTSK, MIN,INDEX,SET(420),ROW,COL,SETUP(50,50)
      REAL PR(65,5,4), TIME, RMIN
      COMMON/MODEL5/IDLMS, PTASK, LTASK, PQTY, VENQTY, VENGRD, TK, DA,
     *SEQ, NUM, PN, PR, PRQTY, NTSK, TOTPM, TOTPL,
     *INIT, MACH, NUMTSK, TIME, TSEQ, SCH, PTLEN
      COMMON /MODEL7/ SETUP,ROW,COL,SET,INDEX,TMIN
C
      IF(NUMTSK.GT.1) GO TO 5
      TSEQ(1)=SCH(1)
      TMIN=0
      RETURN
C
 5    CALL MATRX1
C
C    SETTING UP THE HEURISTIC SO THAT IT WILL USE THE FIRST INIT ORDERS
C    AS THE ORIGINS.
C
      TMIN=30000
      DO 100 P=1,INIT
C
C    INITIALIZING THE TOTAL SET UP TIME COUNTER
C
      TOTTSK=0
      M=P
C
C    SETTING THE PTH ORDER AS THE ORIGIN
C
      TSCH(1)=SCH(M)
C
C    SETTING UP THE LOOPS TO CHECK EVERY ORDER ASSIGNED TO THIS
C    MACHINE
C
      DO 60 I=1,NUMTSK-1
      MIN=10000
C
      DO 55 J=1,NUMTSK
C
C    AVOIDING DIAGONAL ELEMENT OF SET UP TIME MATRIX
C
      IF (SCH(M).EQ.SCH(J)) GO TO 55
C
      DO 45 K=1,I
C
C    CHECKING TO SEE IF CANDIDATE IS ALREADY IN THE CURRENT SEQUENCED
C    LIST
```

```
C
      IF (SCH(J).EQ.TSCH(K)) GO TO 55
 45   CONTINUE
C
C   CHECKING TO SEE IF SET UP TIME OF CANDIDATE IS LESS THAN MINIMUM OF
C   ALL PRIOR CANDIDATES
C
      IF (SETUP(M,J).GT.MIN) GO TO 55
      MIN=SETUP(M,J)
      SQ=J
 55   CONTINUE
C
      TOTTSK=TOTTSK+MIN
      M=SQ
C
C   ASSIGNING BEST CANDIDATE TO CURRENT SEQUENCED LIST
C
      N=I+1
      TSCH(N)=SCH(M)
 60   CONTINUE
C
C   COMPARING CURRENT SEQUENCED LIST WITH PRIOR LISTS
C
      IF (TOTTSK.GE.TMIN) GO TO 100
      TMIN=TOTTSK
      DO 80 L=1,NUMTSK
      TSEQ(L)=TSCH(L)
 80   CONTINUE
C
 100  CONTINUE
      RETURN
      END
```

```
C
C   FEB. 20.1984
C
C   THIS SUBPROGRAM ALONG WITH SUB. MATRX2 DEVELOPS THE SET UP TIME
C   MATRIX TO BE USED IN SUB. TSKSEQ TO DETERMINE THE BEST SEQUENCE
C   FOR THE CURRENT MACHINE.  THIS MATRIX IS CONSTRUCTED USING THE
C   BILL OF MATERIAL OBTAINED FROM PANEL.DAT IN SUB. INCONS.
C
C   AUTHOR: ERIC SCHILD
C
      SUBROUTINE MATRX1
      INTEGER VENQTY(420.5).VENGRD(420.5).TOTPM(6).PTASK.TMIN.
     *IDLMS(18.3.4).LTASK.PQTY.TOTPL(3),PRODID.SHIFT.CNT.
     *MAXTK(3.5).CPTY(4).SEQ(65.5.4),NUM(5.4).PN.PTLEN(4).MACH.
     *NUMTSK.INIT.TSEQ(200).TK.DA,PRQTY(420).NTSK(3.5.6).MACHID.DAY.
     *PNUM(10).PMACH(10).PRQ(420).LFLAG.PRRGRD(6.5).PRRLEN(6.3).
     *PG.PL.NPG.NPL.SETUP(50.50).ROW,COL.SET(420).GRAD.COUNT.SCH(200).
     *INDEX
      REAL PR(65.5,4).TIME.PRATE(420).TOTP,PRTM(420)
      COMMON /MODEL1/ MAXTK,CPTY
      COMMON /MODEL5/ IDLMS,PTASK.LTASK.PQTY,VENQTY.VENGRD.TK.DA,
     *SEQ.NUM.PN,PR,PRQTY,NTSK,TOTPM.TOTPL.INIT.
     *MACH.NUMTSK,TIME,TSEQ.SCH,PTLEN
      COMMON /MODEL6/ PRTM,PRATE.PRQ
      COMMON /MODEL7/ SETUP.ROW.COL.SET.INDEX.TMIN
C
      DO 55 ROW=1.NUMTSK
      DO 50 COL=1,NUMTSK
      SETUP(ROW.COL)=0
      IF(ROW.NE.COL)GO TO 5
C
C   SETTING DIAGONAL ELEMENTS TO INFINITY
C
      SETUP(ROW.COL)=10000
      GO TO 50
5     DO 10 CNT=1.INDEX
      IF(SCH(ROW).EQ.SET(CNT))GO TO 25
10    CONTINUE
      DO 15 CNT=1.INDEX
C
C   DETERMINING WHETHER CURRENT TASK PAIR REQUIRES HIGH SET UP TIME
C
      IF(SCH(COL).NE.SET(CNT))GO TO 15
      SETUP(ROW.COL)=500
      GO TO 25
15    CONTINUE
C
C   DEVELOPING SET UP TIME FOR TASK PAIR BY EVALUATING BILL OF MATERIAL
C   FOR EACH OF THE FOUR TASK GROUPS (FACE, BACK, CNTR. AND XBND)
C
25    IF(VENGRD(SCH(ROW),1).EQ.VENGRD(SCH(COL),1))GO TO 30
      SETUP(ROW,COL)=SETUP(ROW.COL)+30
30    IF(VENGRD(SCH(ROW),2).EQ.VENGRD(SCH(COL),2))GO TO 35
      SETUP(ROW,COL)=SETUP(ROW,COL)+30
35    CALL MATRX2
50    CONTINUE
55    CONTINUE
      RETURN
      END
```

```
C
C   FEB. 20,1984
C
C   THIS SUBPROGRAM ALONG WITH SUB. MATRX1 DEVELOPS THE SET UP TIME
C   MATRIX TO BE USED IN SUB. TSKSEQ TO DETERMINE THE BEST SEQUENCE FOR
C   THE CURRENT MACHINE.  THIS MATRIX IS CONSTRUCTED USING THE BILL OF
C   MATERIAL OBTAINED FROM PANEL.DAT IN SUB. INCONS.
C
C   AUTHOR:  ERIC SCHILD
C
      SUBROUTINE MATRX2
      INTEGER VENQTY(420,5),VENGRD(420,5),TOTPM(6),PTASK,
     *IDLMS(18,3,4),LTASK,PQTY,TOTPL(3),PRODID,SHIFT,TSK,
     *MAXTK(3,5),CPTY(4),SEQ(65,5,4),NUM(5,4),PN,PTLEN(4),MACH,
     *NUMTSK,INIT,TSEQ(200),TK,DA,PRQTY(420),NTSK(3,5,6),MACHID,DAY,
     *PNUM(10),PMACH(10),PRQ(420),LFLAG,PRRGRD(6,5),PRRLEN(6,3),
     *PG,PL,NPG,NPL,GRAD,COUNT,SCH(200),SETUP(50,50),ROW,COL,SET(420),
     *INDEX,TMIN
      REAL PR(65,5,4),TIME,PRATE(420),TOTP,PRTM(420)
      COMMON /MODEL1/ MAXTK,CPTY
      COMMON /MODEL5/ IDLMS,PTASK,LTASK,PQTY,VENQTY,VENGRD,TK,DA,
     *SEQ,NUM,PN,PR,PRQTY,NTSK,TOTPM,TOTPL,INIT,
     *MACH,NUMTSK,TIME,TSEQ,SCH,PTLEN
      COMMON /MODEL6/ PRTM,PRATE,PRQ
      COMMON /MODEL7/ SETUP,ROW,COL,SET,INDEX,TMIN
C
C    DEVELOPING SET UP TIME FOR TASK PAIR BY EVALUATING BILL OF MATERIAL
C    FOR EACH OF THE FOUR TASK GROUPS
C        CNTR: TSK=3,   CNTR OR XBND: TSK=4,   XBND:  TSK=5
C
      DO 20 TSK=3,5
      IF(VENQTY(SCH(ROW),TSK)-VENQTY(SCH(COL),TSK).GE.0)GO TO 10
      IF(VENQTY(SCH(ROW),TSK).GT.0)GO TO 5
      SETUP(ROW,COL)=SETUP(ROW,COL)+30
      GO TO 20
5     IF(VENGRD(SCH(ROW),TSK).EQ.VENGRD(SCH(COL),TSK))GO TO 20
      SETUP(ROW,COL)=SETUP(ROW,COL)+30
      GO TO 20
10    IF(VENQTY(SCH(ROW),TSK)-VENQTY(SCH(COL),TSK))GO TO 15
      IF(VENQTY(SCH(ROW),TSK).EQ.0)GO TO 20
      IF(VENGRD(SCH(ROW),TSK).EQ.VENGRD(SCH(COL),TSK))GO TO 20
      SETUP(ROW,COL)=SETUP(ROW,COL)+30
      GO TO 20
15    IF(VENQTY(SCH(COL),TSK).EQ.0)GO TO 20
      SETUP(ROW,COL)=SETUP(ROW,COL)+30
20    CONTINUE
      RETURN
      END
```

```
C
C     FEB. 21,1984
C
C     THIS SUBPROGRAM WORKS ALONG SIDE PHASE TWO TO
C
C     1) RECALCULATE THE PIECE COUNT FOR EACH ORDER SO THAT
C             THE QUANTITIES ARE ALL EXPRESSED IN A UNIT BASIS
C             (ONE UNIT IS EQUAL TO APPROXIMATELY 150 PIECES OF
C             OF VENEER: THIS WILL VARY BY THE THICKNESS OF THE
C             VENEER)
C
C     2) CONDENSE PAT.DAT SO THAT EACH ORDER IS ONLY GIVEN ONCE.
C
C     3) RESEQUENCE PAT.DAT SO THAT THE ORDERS ARE PLACED IN
C           INCREASING ORDER BASED ON DUE DATES.
C
C     AUTHOR:  ERIC SCHILD
C
      SUBROUTINE RECODE
      INTEGER CODE(3),TYPE,QTY,TIM,THK(7)
      REAL TOTQTY,FCTR(4)
      CALL ASSIGN(2,'SY:PAT.DAT')
      CALL ASSIGN(3,'SY:PAT.DAT')
C
C   UNIT LOAD CONVERSION FACTORS FOR 7/32, 1/6, 1/10, AND 1/12 VENEERS
C
      DATA FCTR/105.,135.,225.,270./
C
C   SINCE THIS VERSION ONLY ALLOWS FOUR THICKNESS CATEGORIES ANY GRADES
C   WHOSE THICKNESS IS OUTSIDE THIS RANGE WILL BE SUBSTITUTED WITH THE
C   THE FOLLOWING THICKNESS CODES
C
      DATA THK/1,2,3,4,2,2,3/
5     READ(2,10,END=25) (CODE(M),M=1,3),QTY,TIM,TYPE
10    FORMAT(T2,3I1,I5,I6,I2)
C
C    ADVANCING ALL DUE DATE TIMES UP 21600 MINUTES
C
      TIM=TIM+21600
C
C   IF THICKNESS CODE OUT OF RANGE, MAKE SUBSTITUTION
C
      CODE(2)=THK(CODE(2))
      TOTQTY=FLOAT(QTY)/FCTR(CODE(2))
      WRITE(3,20) (CODE(M),M=1,3),TOTQTY,TIM,TYPE
20    FORMAT(T2,3I1,F7.2,I6,I2)
      GO TO 5
25    CONTINUE
      CALL CLOSE(2)
      CALL CLOSE(3)
      CALL CONDNS
      RETURN
      END
```

```
C
C     FEB. 21,1984
C
C     THIS SUBPROGRAM IS USED TO CONDENSE PAT.DAT SO THAT EACH ORDER
C     IS ONLY GIVEN ONCE
C
C     AUTHOR:  ERIC SCHILD
C
      SUBROUTINE CONDNS
      INTEGER CODE(500),DCODE(500),TIM(500),
     *TIME(500),NTSK,DTSK(1000),COUNT,TYPE(500),TP(500)
      REAL TOTQTY(500),TQTY(500)
      COMMON /SYSTEM/ DCODE,COUNT,TQTY,TIME,TP
      CALL ASSIGN(2,'SY:PAT.DAT')
      DATA DTSK/1000*0/
      NTSK=1
C
 5    READ(2,10,END=20) CODE(NTSK),TOTQTY(NTSK),TIM(NTSK),
     *TYPE(NTSK)
10    FORMAT(I2,I3,F7.2,I6,I2)
      NTSK=NTSK+1
      GO TO 5
20    CONTINUE
      CALL CLOSE(2)
      NTSK=NTSK-1
C
C   CONDENSING PAT.DAT
C
      K=1
      DO 40 J=1,NTSK
      N=J
C
C    FLAG THAT INDICATES AN ORDER HAS ALREADY BE ELIMINATED
C
      IF(DTSK(N).EQ.1)GO TO 40
      DO 25 I=N+1,NTSK
      M=I
C
C   IF ORDER M IS EQUIVALENT TO ORDER N, THEN CONSOLIDATE OTHERWISE CHECK
C   NEXT ORDER
C
      IF(.NOT.((CODE(M).EQ.CODE(N)).AND.(TIM(M).EQ.TIM(N))
     *.AND.(TYPE(M).EQ.TYPE(N))))GO TO 25
      TOTQTY(N)=TOTQTY(N)+TOTQTY(M)
      DTSK(M)=1
25    CONTINUE
      TIME(K)=TIM(N)
      DCODE(K)=CODE(N)
      TQTY(K)=TOTQTY(N)
      TP(K)=TYPE(N)
      K=K+1
40    CONTINUE
      COUNT=K-1
      CALL ORDER
      RETURN
      END
```

```
C
C   FEB. 21, 1984
C
C   THIS SUBROUTINE IS USED TO RESEQUENCE THE ORDER FILE PAT.DAT
C   SO THAT THE ORDERS ARE PLACED IN INCREASING ORDER BASED ON
C   DUE DATES.
C
C   AUTHOR:  ERIC SCHILD
C
      SUBROUTINE ORDER
      INTEGER DCODE(500),TIME(500),COUNT,CODE1,MIN,
     *DTSK(1000),TYPE,TP(500)
      REAL TQTY(500),QTY1
      COMMON /SYSTEM/ DCODE,COUNT,TQTY,TIME,TP
      CALL ASSIGN(2,'SY:PAT.DAT')
      DATA DTSK/1000*0/
C
C   REORDERING PAT.DAT
C
      DO 40 J=1,COUNT
      MIN=30000
      DO 25 I=1,COUNT
      N=I
C
C   CHECKING TO SEE IF NTH ORDER HAS BE SEQUENCED YET
C
      IF(DTSK(N).EQ.1) GO TO 25
C
C   IF NOT CHECK DUE DATE OF NTH ORDER WITH THE SMALLEST DUE DATE
C   ALREADY FOUND
C
      IF(TIME(N).GT.MIN)GO TO 25
      MIN=TIME(N)
      CODE1=DCODE(N)
      QTY1=TQTY(N)
      TYPE=TP(N)
      M=I
25    CONTINUE
C
C   WRITING REORDERED FILE
C
      WRITE(2,30) CODE1,QTY1,MIN,TYPE
30    FORMAT(I2,I3,F7.2,I6,I2)
      DTSK(M)=1
40    CONTINUE
      CALL CLOSE(2)
      RETURN
      END
```

APPENDIX A.2 VARIABLE DEFINITIONS AND PROGRAM LISTING FOR PHASE TWO

A.2.1 Variable Definition and Listing of Veneer Development Program

AHEAD - Dummy variable use to retain identity of next row of linked storage array.

BEHIND - Dummy variable used to retain identity of previous row of linked storage array.

CODE - Dry veneer code used to locate a veneer grade in the linked inventory array.

DDATE(i) or COMPL - Time the order is due at the ith operation (Spreader, Patcher, or Jointer). i=1,3.

DINVEN(i,j,k,l) - Index code for the ith dry veneer grade, jth dry veneer thickness, kth dry veneer length, lth dry veneer width; used in accessing and storing veneer in the linked inventory array. i=1,11, j=1,4, k=1,5, l=1,3.

DLT(i) - Row number of linked storage array to be deleted. i=1,75.

FGRD(i) - Grade code of ith full width alternative veneer source. i=1,3.

FPAT(i) - Patcher switch indicating whether ith alternative source needs patching. i=1,3.

FRC(i,j) - Veneer development breakdown fraction for the ith column of either PATDEV.DAT or DRYDEV.DAT. i=1,10, j=1,2

FTOTAL - Total number of full width alternative veneer sources (max. of 3).

FWID(i,j) - Width codes that may be used to develop ith alternative veneer source. i=1,3, j=1,4.

GD(i,j) - Veneer grade of the ith column of the jth development file, PATDEV.DAT or DRYDEV.DAT. i=1,10, j=1,2.

GINVEN(i,j,k,l,m) - Index code for the ith green veneer peel class, jth green veneer grade, kth green veneer thickness, lth green veneer length, mth green veneer width; used in accessing entries from DRYDEV.DAT. i=1,2, j=1,6, k=1,4, l=1,5, m=1,3.

GRA(i,j) - Grade codes of full width veneer grades that will be used to fill veneer orders when developing ith alternative veneer source on patcher operation, i=1,3, j=1,2.

GRADE(i,j) - Character string designating the ith dry veneer grades. i=1,11, j=1,4.

GRD - Veneer grade code.

HEAD(i) - Head pointer for list i. i=1,660.

IHEAD - Dummy variable used to retain identity of HEAD(i).

ITAIL - Dummy variable used to retain identity of TAIL(i).

JRATE(i,j) - Production rate for jointing a veneer grade with thickness i, and length j. i=1,4, j=1,5.

LEN - Veneer length code.

LENGTH(i) - Veneer length.

LINKPR(i) - Predecessor link for row i of linked storage array. i=1,1500.

LINKSR(i) - Successor link for row i of linked storage array. i=1,1500.

LSIZE(i) - Number of records in list i. i=1,660.

MASTER(i) - Linked inventory array. i=1,1500, j=1,3

NAR - Next available row in linked storage array.

NGRD(i) - Total number of full width veneer grades that will be used to fill veneer order when developing alternative veneer source on patcher operation. i=1,3.

NPATCH(i,j,k,l) - Index code for the ith dry veneer grade, jth dry veneer thickness, kth dry veneer length, and lth dry veneer width; used in accessing entries from PATDEV.DAT. i=1,9, j=1,4, k=1,4, l=1,3.

NROWS - Number of rows in linked storage array to be deleted.

ORIQTY - Order quantity.

P - Current operation.

PD - Index code used to designate which development file is being used (PATDEV.DAT or, DRYDEV.DAT).

PRATE(i) - Production rate for the ith operation. i=1,3.

PROW(i) - List of development codes indicating the tasks used to create a dry veneer source. i=1,26.

QTY - Order quantity.

ROW - Dummy variable used to retain identity of a row of linked storage array.

SS1 - Veneer grade code.

SS2 - Veneer width code.

TAIL(i) - Tail pointer for list i. i=1,660.

THICK(i,j) - Character string designating the ith dry veneer thickness. i=1,11, j=1,4.

THK - Veneer thickness code.

TRNSFR(i) - Transfer array assists in storing the veneer quantity, event time, and development code in the linked storage array. i=1,3.

TROW - Development code indicating the task used to create a dry veneer source.

TYP - Veneer type (face, back, center, or crossband).

VCODE - Index code for accessing entries from VENDEV.DAT.

VENDV(i,j,k) - Index code for the ith dry veneer grade, jth dry veneer thickness, and kth dry veneer length; used in accessing entries from VENDEV.DAT.

WID(i,j) - Character string designating the ith dry veneer width. i=1,11, j=1,4.

XGRD(i) - Grade of ith narrow width alternative veneer source. i=1,3.

XTOTAL - Total number of narrow width alternative veneer sources (max. of 3).

XWID(i,j) - Width codes that may be used to develop ith alternative veneer source. i=1,3, j=1,4.

```
C
C   FEB. 21,1984
C
C  THIS PROGRAM READS THE ENTRIES FROM THE MASTER PRODUCTION SCHEDULE
C  CREATED IN PHASE ONE SO THAT EACH REQUIRED VENEER GRADE CAN BE
C  DEVELOPED BY CREATING VENEER TASKS ON ONE OR ALL THREE OF THE UP-
C  STREAM OPERATIONS.
C
C   AUTHOR:  ERIC SCHILD
C
      PROGRAM ALLOC
      INTEGER NPATCH(9.4.4.3),FTOTAL.FGRD(3).GINVEN(2.6.4.5.3).
     *FPAT(3),NGRD(3).GRA(3.2).FWID(3,4).XTOTAL.XGRD(3),XWID(3.4),
     *GD(10.2).DINVEN(11.4.5,3),COMPL.GRD.THK.LEN.TYP.DDATE(3).
     *SS1.SS2.CODE.PD.P,TAIL(660),HEAD(660).LINKPR(1500).NAR.ROW.
     *LINKSR(1500).MASTER(1500.3).AHEAD,BEHIND,IHEAD.ITAIL.ITEM.SIZE.
     *NROWS.DLT(75).LSIZE(660).TRNSFR(3).PROW(26).NM.TROW.LN.
     *LENGTH(11).VENDV(8,4,4).CNT.CNTR.VCODE
      REAL QTY.ORIQTY,FRC(11.2).PRATE(3).JRATE(4.4)
      LOGICAL FILLED
      BYTE GRADE(11,4).THICK(11.4).WID(11.2)
      COMMON /MODEL7/ NPATCH,GINVEN,VENDV,DINVEN,JRATE.FTOTAL.FGRD.
     *FPAT.FWID.NGRD.GD,LN,GRA,XTOTAL.XGRD.XWID.CNT.CNTR.VCODE
      COMMON /MODEL8/ GRD,THK,LEN.TYP.DDATE.ORIQTY.SS1,SS2.CODE.QTY,
     *FRC.PRATE.PD.P.COMPL,NM.PROW.TROW.FILLED.GRADE.THICK.LENGTH.
     *WID
      COMMON /MODEL9/ TAIL.HEAD,LINKPR.MASTER.NAR,ROW,AHEAD.BEHIND.
     *IHEAD.ITAIL,ITEM.DLT.SIZE,LINKSR.TRNSFR,LSIZE
C
      CALL RCODE1
C
C  OPENING THE MASTER PRODUCTION FILE
C
      CALL ASSIGN(1,'SY:PAT.DAT')
C
C   OPENING THE PATCHER DEVELOPMENT FILE, THE DRYER DEVELOPMENT FILE.
C   AND THE VENEER DEVELOPMENT FILE
C
      OPEN(UNIT=2.TYPE='OLD'.ACCESS='DIRECT'.NAME='SY:PATDEV.DAT'.
     *RECORDSIZE=15.READONLY)
      OPEN(UNIT=3.TYPE='OLD',ACCESS='DIRECT',NAME='SY:DRYDEV.DAT'.
     *RECORDSIZE=15.READONLY)
      OPEN(UNIT=4.TYPE='OLD',ACCESS='DIRECT'.NAME='SY:VENDEV.DAT'.
     *RECORDSIZE=25.READONLY)
C
      CALL ASSIGN(7,'SY:PAT1.DAT')
C
C  READING AN ORDER FROM THE MASTER PRODUCTION FILE
C
 5    READ(1.10.END=25) GRD.THK.LEN.ORIQTY.DDATE(1).TYP
10    FORMAT(I2,3I1.F7.2.I6.I2)
C
      TYPE*.DDATE(1).ORIQTY
      FILLED=.FALSE.
      CALL INICNT
      IF(FILLED) GO TO 5
      IF(TYP.GT.3)GO TO 20
      CALL BCKPS1
      IF(FILLED) GO TO 5
20    CALL DRYER
```

```
      GO TO 5
25    CONTINUE
C
C  FINAL CALL TO THE SLAVE PROGRAM
C
      CALL TRNSMT(-1,0.0,0,0.0,0.0,0,0,0.)
      CALL CLOSE(1)
      CALL CLOSE(2)
      CALL CLOSE(3)
      CALL CLOSE(4)
      CALL CLOSE(7)
      STOP
      END
```

```
C
C   APRIL 30, 1984
C
C   THIS SUBROUTINE IS USED TO SEND THE NECESSARY DATA TO
C   THE SLAVE ROUTINE SO THAT THE CURRENT TASK CAN BE SCHEDULED.
C
C   AUTHOR:  ERIC SCHILD
C
      SUBROUTINE TRNSMT(FLAG,OPER,PRTM,GRD,THK,LEN,WID,CODE,DUE,TROW,
     *PRTE,NM,PROW)
      INTEGER FLAG,OPER,PRTM,GRD,THK,LEN,WID,CODE,DUE,TM,J(13),K(15),
     *PROW(26),TROW,SIZE
      REAL PRTE
C
      DATA FLE/5RPHAS3/
      WRITE(6,5) GRD,THK,LEN,WID,CODE,(PROW(M),M=1,NM)
5     FORMAT(T2,'TRNSMT CODE-',I2,3I1,' CODE-',I5,' SOURCE-',26I5)
      J(1)=FLAG
      J(2)=OPER
      J(3)=PRTM
      J(4)=GRD
      J(5)=THK
      J(6)=LEN
      J(7)=WID
      J(8)=CODE
      J(9)=DUE
      J(10)=IFIX(PRTE*100000.)
      J(11)=NM
      CALL CLREF(33)
      CALL SEND(FLE,J,33)
      CALL WAITEF(37)
      CALL CLREF(37)
      SIZE=NINT(FLOAT(NM)/13.)
      N=1
      DO 20 I=1,SIZE
      DO 10 M=1,13
      J(M)=PROW(N)
      N=N+1
10    CONTINUE
      CALL CLREF(33)
      CALL SEND(FLE,J,33)
      CALL WAITEF(37)
      CALL CLREF(37)
20    CONTINUE
C
C    IF FLAG IS LESS THAN 0, TERMINATE PROGRAM
C
      IF(FLAG.LT.0)RETURN
C
      CALL WAITEF(35)
      CALL RECEIV(FLE,K)
      CALL CLREF(35)
      TROW=K(3)
      WRITE(6,25) GRD,THK,LEN,WID,CODE,TROW
25    FORMAT(T2,'TRNSMT RECEIVE-',I2,3I1,' CODE-',I5,' TASK-',I5)
      RETURN
      END
```

```
C
C     FEB. 21,1984
C
C     THIS SUBPROGRAM IS USED TO SET UP THE 4(OR 5) DIGIT CODES THAT WILL
C     BE USED PERIODICALLY THROUGHOUT THE REMAINDER OF THIS PROGRAM TO
C     ACCESS ENTRIES FROM THE PATCHER DEVELOPMENT FILE. THE DRYER DEVELOP-
C     MENT FILE. THE VENEER DEVELOPMENT FILE. AND THE INVENTORY FILE.  IT
C     IS ALSO USED TO READ IN THE JOINTER PRODUCTION RATES.
C
C     AUTHOR:  ERIC SCHILD
C
      SUBROUTINE RCODE1
      INTEGER NPATCH(9.4.4,3).FTOTAL.FGRD(3).GINVEN(2,6,4.5.3).
     *FPAT(3).NGRD(3).GRA(3.2).FWID(3.4).XTOTAL.XGRD(3).XWID(3,4).
     *GD(10.2).DINVEN(11.4.5.3).COMPL.GRD.THK.LEN.TYP.DDATE(3).
     *SS1,SS2.CODE.PD.P.TAIL(660).HEAD(660).LINKPR(1500).NAR.ROW.
     *LINKSR(1500).MASTER(1500.3).AHEAD.BEHIND.IHEAD.ITAIL.ITEM.SIZE.
     *NROWS.DLT(75).LSIZE(660).TRNSFR(3).PROW(26).NM.TROW.LN.
     *LENGTH(11).VENDV(8.4.4).CNT.G.T.L.W.CNTR.VCODE
      REAL QTY.RATE.ORIQTY.FRC(11.2).PRATE(3).JRATE(4.4)
      LOGICAL FILLED
      BYTE GRADE(11.4),THICK(11.4),WID(11.2)
      COMMON /MODEL7/ NPATCH.GINVEN,VENDV.DINVEN.JRATE.FTOTAL.FGRD.
     *FPAT.FWID.NGRD.GD.LN.GRA.XTOTAL.XGRD.XWID.CNT.CNTR.VCODE
      COMMON /MODEL8/ GRD,THK,LEN,TYP,DDATE.ORIQTY.SS1.SS2.CODE.QTY.
     *FRC.PRATE.PD.P.COMPL.NM.PROW.TROW.FILLED.GRADE.THICK.LENGTH.
     *WID
      COMMON /MODEL9/ TAIL.HEAD.LINKPR.MASTER.NAR.ROW.AHEAD.BEHIND.
     *IHEAD.ITAIL.ITEM.DLT.SIZE,LINKSR.TRNSFR.LSIZE
C
      CALL ASSIGN(1.'SY:PATCOD.DAT')
      CALL ASSIGN(2.'SY:INVCOD.DAT')
      CALL ASSIGN(3.'SY:DRYCOD.DAT')
      CALL ASSIGN(4.'SY:VENCOD.DAT')
C
C    CREATING THE CODING SCHEMES FOR THE PATCHER DEVELOPMENT FILE.
C    THE DRYER DEVELOPMENT FILE , THE VENEER DEVELOPMENT FILE. AND THE
C    INVENTORY FILE.
C
      DO 55 J=1,4
      M=J
      INDEX=1
      GO TO (5.15.25.35) M
5     READ(M,10,END=50) G.T.L,W
10    FORMAT(T2.I2.3I1)
      NPATCH(G.T.L.W)=INDEX
      GO TO 45
15    READ(M.20.END=50) G.T.L,W
20    FORMAT(T2.I2.3I1)
      DINVEN(G.T.L.W)=INDEX
      GO TO 45
25    READ(M.30.END=50) N.G.T.L,W
30    FORMAT(T2.2I2.3I1)
      GINVEN(N.G.T,L.W)=INDEX
      GO TO 45
35    READ(M.40.END=50) G.T.L
40    FORMAT(T2.3I1)
      VENDV(G.T.L)=INDEX
45    INDEX=INDEX+1
      GO TO(5.15.25.35) M
```

```
50    CONTINUE
55    CONTINUE
      CALL CLOSE(1)
      CALL CLOSE(2)
      CALL CLOSE(3)
      CALL CLOSE(4)
C
      CALL ASSIGN(1,'SY:GRADE.DAT')
      INDEX=1
C
C   READING THE FILE THAT WILL BE USED TO INTERPRET THE COLUMNS
C   OF THE PATCHER AND DRYER DEVELOPMENT FILES.
C
60    READ(1,65,END=70) (GD(INDEX,M),M=1,2)
65    FORMAT(T2,2I2)
      INDEX=INDEX+1
      GO TO 60
70    CONTINUE
      CALL CLOSE(1)
C
C   READING PRODUCTION RATES FOR THE JOINTER OPERATION. NOTE THAT
C   T=VENEER THICKNESS AND L=VENEER LENGTH.
C
      CALL ASSIGN(1,'SY:JRATE.DAT')
75    READ(1,80,END=85) T,L,RATE
80    FORMAT(2I2,F7.5)
      JRATE(T,L)=RATE
      GO TO 75
85    CONTINUE
      CALL CLOSE(1)
      CALL RCODE2
      RETURN
      END
```

```
C
C   FEB. 21,1984
C
C   THIS SUBPROGRAM CONSTRUCTS THE INITIAL DRY INVENTORY THAT THE USER
C   SPECIFIES IN THE DRY INVENTORY FILE.
C
C   AUTHOR:  ERIC SCHILD
C
      SUBROUTINE RCODE2
      INTEGER NPATCH(9,4,4,3),FTOTAL,FGRD(3),GINVEN(2,6,4,5,3),
     *FPAT(3),NGRD(3),GRA(3,2),FWID(3,4),XTOTAL,XGRD(3),XWID(3,4),
     *GD(10,2),DINVEN(11,4,5,3),COMPL,GRD,THK,LEN,TYP,DDATE(3),
     *SS1,SS2,CODE,PD,P,TAIL(660),HEAD(660),LINKPR(1500),NAR,ROW,
     *LINKSR(1500),MASTER(1500,3),AHEAD,BEHIND,IHEAD,ITAIL,ITEM,SIZE,
     *NROWS,DLT(75),LSIZE(660),TRNSFR(3),PROW(26),NM,TROW,LN,
     *LENGTH(11),VENDV(8,4,4),CNT,G,T,W,QUNTY,TIME,CNTR,VCODE
      REAL QTY,ORIQTY,FRC(11,2),PRATE(3),JRATE(4,4)
      LOGICAL FILLED
      BYTE GRADE(11,4),THICK(11,4),WID(11,2)
      COMMON /MODEL7/ NPATCH,GINVEN,VENDV,DINVEN,JRATE,FTOTAL,FGRD,
     *FPAT,FWID,NGRD,GD,LN,GRA,XTOTAL,XGRD,XWID,CNT,CNTR,VCODE
      COMMON /MODEL8/ GRD,THK,LEN,TYP,DDATE,ORIQTY,SS1,SS2,CODE,QTY,
     *FRC,PRATE,PD,P,COMPL,NM,PROW,TROW,FILLED,GRADE,THICK,LENGTH,
     *WID
      COMMON /MODEL9/ TAIL,HEAD,LINKPR,MASTER,NAR,ROW,AHEAD,BEHIND,
     *IHEAD,ITAIL,ITEM,DLT,SIZE,LINKSR,TRNSFR,LSIZE
      CALL ASSIGN(1,'SY:DVEN.DAT')
C
C  THE FOLLOWING BYTE ARRAYS WILL BE USED IN SUBROUTINE DRYER TO
C  TRANSLATE THE NUMERIC IDENTIFICATION CODE INTO AN UNDERSTANDABLE
C  CHARACTER STRING.
C
      INDEX=1
5     READ(1,10,END=15) (GRADE(INDEX,M),M=1,4),(THICK(INDEX,M),M=1,4),
     *LENGTH(INDEX),(WID(INDEX,M),M=1,2)
10    FORMAT(T2,4A1,4A1,I2,2A1)
      INDEX=INDEX+1
      GO TO 5
15    CONTINUE
      CALL CLOSE(1)
C
C  INITIALIZING THE LINKED LIST ARRAY THAT WILL BE USED TO STORE THE
C  ON-HAND AND ON-ORDER DRY INVENTORY.
C
      CALL INITK1
C
C   SETTING UP THE INITIAL DRY INVENTORY LEVEL AS PRESCRIBED BY THE USER
C
      CALL ASSIGN(1,'SY:DRY.DAT')
20    READ(1,25,END=30) G,T,LN,W,QUNTY,TIME
25    FORMAT(T2,I2,3I1,2I5)
C
      TRNSFR(1)=TIME
      TRNSFR(2)=QUNTY*100
      TRNSFR(3)=-1
      INDEX=DINVEN(G,T,LN,W)
C
C   STORING DRY INVENTORY IN THE LINKED STORAGE ARRAY
C
      CALL FILE(INDEX)
```

```
      GO TO 20
30    CONTINUE
      CALL CLOSE(1)
      RETURN
      END
```

```
C
C    FEB. 21.1984
C
C    THIS SUBPROGRAM IS USED TO INITIALIZE THE LINKED STORAGE STRUCTURE
C    FOR FILING DRY INVENTORY.  IT IS IDENTICAL TO THE ROUTINE USED IN
C    THE SIMULATION LANGUAGE SIMLIB EXCEPT FOR A FEW MINOR MODIFICATIONS.
C
      SUBROUTINE INITK1
      INTEGER NPATCH(9.4.4.3).FTOTAL.FGRD(3).GINVEN(2.6.4.5.3).
     *FPAT(3),NGRD(3).GRA(3.2).FWID(3,4).XTOTAL.XGRD(3).XWID(3.4).
     *GD(10.2).DINVEN(11.4.5.3).COMPL.GRD.THK.LEN,TYP.DDATE(3).
     *SS1.SS2,CODE.PD.P.TAIL(660).HEAD(660).LINKPR(1500).NAR.ROW.
     *LINKSR(1500).MASTER(1500,3).AHEAD.BEHIND.IHEAD.ITAIL.ITEM.SIZE.
     *NROWS.DLT(75),LSIZE(660).TRNSFR(3).PROW(26).NM.TROW.LN.
     *LENGTH(11).VENDV(8,4.4).CNT,CNTR,VCODE
      REAL QTY.ORIQTY.FRC(11.2).PRATE(3).JRATE(4,4)
      LOGICAL FILLED
      BYTE GRADE(11,4).THICK(11.4).WID(11.2)
      COMMON /MODEL7/ NPATCH,GINVEN,VENDV.DINVEN.JRATE.FTOTAL.FGRD.
     *FPAT.FWID.NGRD.GD.LN.GRA.XTOTAL.XGRD.XWID.CNT.CNTR.VCODE
      COMMON /MODEL8/ GRD.THK,LEN.TYP.DDATE,ORIQTY.SS1.SS2,CODE.QTY.
     *FRC.PRATE.PD.P.COMPL.NM.PROW.TROW.FILLED.GRADE.THICK.LENGTH.
     *WID
      COMMON /MODEL9/ TAIL.HEAD.LINKPR.MASTER.NAR.ROW.AHEAD.BEHIND.
     *IHEAD.ITAIL.ITEM.DLT.SIZE,LINKSR.TRNSFR.LSIZE
C
C   INITIALIZING LINKS
C
      DO 10 ROW=1.1500
      LINKPR(ROW)=0
      LINKSR(ROW)=ROW+1
10    CONTINUE
      LINKSR(1500)=0
C
C    INITIALIZING LIST OF ATTRIBUTES
C
      DO 20 LIST=1.660
      HEAD(LIST)=0
      TAIL(LIST)=0
      LSIZE(LIST)=0
20    CONTINUE
      NAR=1
      RETURN
      END
```

```
C
C   FEB. 21,1984
C
C   THIS SUBPROGRAM IS THE STORAGE DEVISE USED TO STORE THE NEWLY
C   GENERATED DRY INVENTORY IN THE MASTER ARRAY.  AS WAS INDICATED
C   IN SUB. INITLK, THIS ROUTINE IS IDENTICAL TO SUB. FILE USED IN
C   SIMLIB EXCEPT ONLY ONE STORING OPTION IS ALLOWED.  THIS OPTION
C   RANKS THE LIST IN DECREASING ORDER BASED ON EVENT TIME (TIME
C   VENEER WAS PLACED IN INVENTORY).
C
      SUBROUTINE FILE(LIST)
      INTEGER NPATCH(9,4,4,3),FTOTAL,FGRD(3),GINVEN(2,6,4,5,3),
     *FPAT(3),NGRD(3),GRA(3,2),FWID(3,4),XTOTAL,XGRD(3),XWID(3,4),
     *GD(10,2),DINVEN(11,4,5,3),COMPL,GRD,THK,LEN,TYP,DDATE(3),
     *SS1,SS2,CODE,PD,P,TAIL(660),HEAD(660),LINKPR(1500),NAR,ROW,
     *LINKSR(1500),MASTER(1500,3),AHEAD,BEHIND,IHEAD,ITAIL,ITEM,SIZE,
     *NROWS,DLT(75),LSIZE(660),TRNSFR(3),PROW(26),NM,TROW,LN,
     *LENGTH(11),VENDV(8,4,4),CNT,LIST,CNTR,VCODE
      REAL QTY,ORIQTY,FRC(11,2),PRATE(3),JRATE(4,4)
      LOGICAL FILLED
      BYTE GRADE(11,4),THICK(11,4),WID(11,2)
      COMMON /MODEL7/ NPATCH,GINVEN,VENDV,DINVEN,JRATE,FTOTAL,FGRD,
     *FPAT,FWID,NGRD,GD,LN,GRA,XTOTAL,XGRD,XWID,CNT,CNTR,VCODE
      COMMON /MODEL8/ GRD,THK,LEN,TYP,DDATE,ORIQTY,SS1,SS2,CODE,QTY,
     *FRC,PRATE,PD,P,COMPL,NM,PROW,TROW,FILLED,GRADE,THICK,LENGTH,
     *WID
      COMMON /MODEL9/ TAIL,HEAD,LINKPR,MASTER,NAR,ROW,AHEAD,BEHIND,
     *IHEAD,ITAIL,ITEM,DLT,SIZE,LINKSR,TRNSFR,LSIZE
C
C   IF THE MASTER STORAGE ARRAY IS FULL, STOP
C
      IF(NAR.GT.0) GO TO 20
      WRITE(5,10)
10    FORMAT(5X,'MASTER STORAGE ARRAY OVERFLOW')
      STOP
C
C   IF THE LIST OF VALUES IS IMPROPER, STOP
C
20    IF((LIST.GE.1).AND.(LIST.LE.660)) GO TO 40
      WRITE(5,30) LIST
30    FORMAT(T2,I10,' IS AN IMPROPER VALUE FOR FILE LIST')
      STOP
C
C   INCREMENT THE LIST SIZE
C
40    LSIZE(LIST)=LSIZE(LIST)+1
C
C   IF THIS IS NOT THE FIRST RECORD IN THIS LIST, CONTINUE
C
130   IF(LSIZE(LIST).EQ.1) GO TO 400
C
C   SEARCH THE LIST FOR PROPER LOCATION
C
      ROW=HEAD(LIST)
C
C   RANKING THE LIST IN DECREASING ORDER
C
140   IF(TRNSFR(1).LT.MASTER(ROW,1)) GO TO 160
      IF((TRNSFR(1).EQ.MASTER(ROW,1)).AND.(TRNSFR(3).EQ.MASTER
     *(ROW,3))) GO TO 600
```

```
C
C   THE CORRECT LOCATION HAS BEEN FOUND.  INSERT BEFORE THE LAST
C   RECORD EXAMINED.
C
150   IF(ROW.EQ.HEAD(LIST)) GO TO 300
C
C   INSERT IN THE PROPER LOCATION BETWEEN THE PRECEEDING AND SUCCEEDING
C   RECORDS
C
      AHEAD=LINKSR(BEHIND)
      ROW=NAR
      NAR=LINKSR(ROW)
      IF(NAR.GT.0) LINKPR(NAR)=0
      LINKPR(ROW)=BEHIND
      LINKSR(BEHIND)=ROW
      LINKPR(AHEAD)=ROW
      LINKSR(ROW)=AHEAD
C
C   GO TO TRANSFER THE DATA
C
      GO TO 500
C
C   CONTINUE SEARCHING, CONSIDER THE NEXT ROW
C
160   BEHIND=ROW
      ROW=LINKSR(BEHIND)
C
C   IF THE LAST ROW CONSIDERED WAS NOT THE TAIL OF THE LIST, CONTINUE
C
      IF(TAIL(LIST).NE.BEHIND) GO TO 140
C
C   INSERT AFTER THE LAST RECORD IN THE LIST
C
200   ROW=NAR
      NAR=LINKSR(ROW)
      IF(NAR.GT.0) LINKPR(NAR)=0
      ITAIL=TAIL(LIST)
      LINKPR(ROW)=ITAIL
      LINKSR(ITAIL)=ROW
      LINKSR(ROW)=0
      TAIL(LIST)=ROW
C
C   GO TO TRANSFER THE DATA
C
      GO TO 500
C
C   INSERT BEFORE THE FIRST RECORD IN THE LIST
C
300   ROW=NAR
      NAR=LINKSR(ROW)
      IF(NAR.GT.0) LINKPR(NAR)=0
      IHEAD=HEAD(LIST)
      LINKPR(IHEAD)=ROW
      LINKSR(ROW)=IHEAD
      LINKPR(ROW)=0
      HEAD(LIST)=ROW
C
C   GO TO TRANSFER THE DATA
C
      GO TO 500
```

```
C
C   INSERT THE FIRST RECORD IN THE LIST
C
400   ROW=NAR
      NAR=LINKSR(ROW)
      IF(NAR.GT.0) LINKPR(NAR)=0
      LINKSR(ROW)=0
      HEAD(LIST)=ROW
      TAIL(LIST)=ROW
C
C   TRANSFER THE DATA
C
500   DO 510 ITEM=1,3
      MASTER(ROW,ITEM)=TRNSFR(ITEM)
510   CONTINUE
      RETURN
C
C   COMBINE DATA WITH CURRENT QUANTITY IN MASTER ARRAY
C
600   MASTER(ROW,2)=MASTER(ROW,2)+TRNSFR(2)
      LSIZE(LIST)=LSIZE(LIST)-1
      RETURN
      END
```

```
C
C     FEB. 21,1984
C
C     THIS SUBROUTINE MAKES THE INITIAL CHECK OF DRY INVENTORY FOR THE
C     DESIRED GRADE.  IF THE CORRECT GRADE IS FOUND, THEN THE VENEER IS
C     REMOVED FROM INVENTORY AND THE AMOUNT FOUND IS APPLIED AGAINST THE
C     CURRENT NEEDS.  IF THE CORRECT GRADE IS NOT FOUND OR THERE ISN'T
C     ENOUGH TO FILL THE CURRENT ORDER, THEN OTHER SOURCES ARE EVALUATED.
C
C     AUTHOR:  ERIC SCHILD
C
      SUBROUTINE INICNT
      INTEGER NPATCH(9,4,4,3),FTOTAL,FGRD(3),GINVEN(2,6,4,5,3),
     *FPAT(3),NGRD(3),GRA(3,2),FWID(3,4),XTOTAL,XGRD(3),XWID(3,4),
     *GD(10,2),DINVEN(11,4,5,3),COMPL,GRD,THK,LEN,TYP,DDATE(3),
     *SS1,SS2,CODE,PD,P,TAIL(660),HEAD(660),LINKPR(1500),NAR,ROW,
     *LINKSR(1500),MASTER(1500,3),AHEAD,BEHIND,IHEAD,ITAIL,ITEM,SIZE,
     *NROWS,DLT(75),LSIZE(660),TRNSFR(3),PROW(26),NM,TROW,LN,
     *LENGTH(11),CD,NT,NR,VENDV(8,4,4),CNT,BD,CNTR,VCODE,DUM(28)
      REAL QTY,ORIQTY,FRC(11,2),PRATE(3),JRATE(4,4)
      LOGICAL FILLED
      BYTE GRADE(11,4),THICK(11,4),WID(11,2)
      COMMON /MODEL7/ NPATCH,GINVEN,VENDV,DINVEN,JRATE,FTOTAL,FGRD,
     *FPAT,FWID,NGRD,GD,LN,GRA,XTOTAL,XGRD,XWID,CNT,CNTR,VCODE
      COMMON /MODEL8/ GRD,THK,LEN,TYP,DDATE,ORIQTY,SS1,SS2,CODE,QTY,
     *FRC,PRATE,PD,P,COMPL,NM,PROW,TROW,FILLED,GRADE,THICK,LENGTH,
     *WID
      COMMON /MODEL9/ TAIL,HEAD,LINKPR,MASTER,NAR,ROW,AHEAD,BEHIND,
     *IHEAD,ITAIL,ITEM,DLT,SIZE,LINKSR,TRNSFR,LSIZE .
C
C     IF THE ORDER CONSIST OF ACQUIRING FULL WIDTH VENEER, CONTINUE
C
      IF(TYP.GT.3) GO TO 10
C
C     4-DIGIT CODE USED IN LOCATING THE CORRECT DRY INVENTORY IN THE
C     LINKED ARRAY.
C
      SS2=1
      LN=LEN
      CODE=DINVEN(GRD,THK,LEN,SS2)
C
C     IF LIST IS EMPTY, RETURN
C
      IF(LSIZE(CODE).EQ.0)RETURN
      QTY=ORIQTY
      CALL NFIND(DDATE(1),CODE,ORIQTY)
      NT=1
      NR=0
      IF(ORIQTY.GE.QTY)RETURN
      QTY=QTY-ORIQTY
      CD=THK*100+LEN*10+SS2
      WRITE(7,5) DDATE(1),QTY,CODE,CD,NT,NR
5     FORMAT(I2,I6,F6.2,2I4,2I2)
      IF(ORIQTY.GT.0)RETURN
      FILLED=.TRUE.
      RETURN
C
C     ORDER CONSISTS OF ACQUIRING NARROW WIDTH VENEER
C
C     READING IN ALTERNATIVE SOURCES FROM THE VENEER DEVELOPMENT FILE
```

```
C
10    VCODE=VENDV(GRD,THK,LEN)
      READ(4'VCODE) (DUM(M),M=1,28),XTOTAL,(XGRD(M),M=1,3),
     *((XWID(N,M),M=1,4),N=1,3)
C
C   ENUMERATING ALL OF THE POSSIBLE ALTERNATIVE SOURCES
C
      BD=XTOTAL
      DO 20 CNT=1,BD
C
      SS1=XGRD(CNT)
      CALL INIXBD
      IF(FILLED)RETURN
20    CONTINUE
      GRD=XGRD(1)
      RETURN
      END
```

```
C
C   FEB. 21,1984
C
C  THIS SUBPROGRAM SEARCHES DRY INVENTORY (LINKED STORAGE ARRAY) FOR THE
C  DESIRED VENEER.  IF VENEER IS FOUND THAT MEETS THE PRESCRIBED DUE
C  DATE. THEN SUBROUTINE DELETE IS CALLED TO PLACE THE ENTRY THAT THIS
C  VENEER OCCUPIED INTO THE NAS (NEXT AVAILABLE SPACE) FILE.  NOTE THAT
C  THIS ROUTINE IS NOT PART OF THE SIMLIB LIBRARY.
C
C   AUTHOR:  ERIC SCHILD
C
      SUBROUTINE NFIND(VALUE1,LIST,VALUE2)
      INTEGER NPATCH(9,4,4,3),FTOTAL,FGRD(3),GINVEN(2,6,4,5,3),
     *FPAT(3),NGRD(3),GRA(3,2),FWID(3,4),XTOTAL,XGRD(3),XWID(3,4),
     *GD(10,2),DINVEN(11,4,5,3),COMPL,GRD,THK,LEN,TYP,DDATE(3),
     *SS1,SS2,CODE,PD,P,TAIL(660),HEAD(660),LINKPR(1500),NAR,ROW,
     *LINKSR(1500),MASTER(1500,3),AHEAD,BEHIND,IHEAD,ITAIL,ITEM,SIZE,
     *NROWS,DLT(75),LSIZE(660),TRNSFR(3),PROW(26),NM,TROW,LN,
     *LENGTH(11),VENDV(8,4,4),CNT,VALUE1,LIST,CNTR,VCODE
      REAL QTY,TQTY,VALUE2,ORIQTY,FRC(11,2),PRATE(3),JRATE(4,4)
      LOGICAL FILLED
      BYTE GRADE(11,4),THICK(11,4),WID(11,2)
      COMMON /MODEL7/ NPATCH,GINVEN,VENDV,DINVEN,JRATE,FTOTAL,FGRD,
     *FPAT,FWID,NGRD,GD,LN,GRA,XTOTAL,XGRD,XWID,CNT,CNTR,VCODE
      COMMON /MODEL8/ GRD,THK,LEN,TYP,DDATE,ORIQTY,SS1,SS2,CODE,QTY,
     *FRC,PRATE,PD,P,COMPL,NM,PROW,TROW,FILLED,GRADE,THICK,LENGTH,
     *WID
      COMMON /MODEL9/ TAIL,HEAD,LINKPR,MASTER,NAR,ROW,AHEAD,BEHIND,
     *IHEAD,ITAIL,ITEM,DLT,SIZE,LINKSR,TRNSFR,LSIZE
C
C   STARTING AT BOTTOM OF LIST
C
      ROW=TAIL(LIST)
      J=0
      NM=0
      DO 30 I=1,LSIZE(LIST)
C
C   IF THE ENTRY TIME IS GREATER THAN THE DESIRED TIME, THEN THE STOCK
C   IS NOT AVAILABLE
C
      IF(MASTER(ROW,1).GT.VALUE1.OR.VALUE2.EQ.0.) GO TO 35
C
C   IF THE ENTRY QUANTITY IS GREATER THAN THE DESIRED QUANTITY. THEN THE
C   ORDER CAN BE FILLED
C
      TQTY=FLOAT(MASTER(ROW,2))/100.
      IF(TQTY.GT.VALUE2) GO TO 15
      VALUE2=VALUE2-TQTY
      J=J+1
      NM=NM+1
      PROW(NM)=MASTER(ROW,3)
      DLT(J)=ROW
      ROW=LINKPR(ROW)
      GO TO 30
C
C   ENTRY QUANTITY WAS LESS THAN DESIRED QUANTITY
C
15    TQTY=TQTY-VALUE2
      MASTER(ROW,2)=NINT(TQTY*100)
      VALUE2=0.
```

```
      NM=NM+1
      PROW(NM)=MASTER(ROW,3)
      GO TO 35
30    CONTINUE
C
C   IF NO INVENTORY WAS FOUND, RETURN
C
35    IF(J.EQ.0)RETURN
      CALL DELETE(J,LIST)
      RETURN
      END
```

```
C
C    FEB. 21.1984
C
C    THIS SUBPROGRAM IS USED IN CONJUNCTION WITH SUB. NFIND TO PLACE
C    THE CONTENTS OF AN ENTRY OF THE MASTER STORAGE ARRAY INTO THE NAS
C    FILE.  NOTE THAT THIS SUBROUTINE IS NOT PART OF THE SIMLIB LIBRARY.
C
C    AUTHOR:  ERIC SCHILD
C
      SUBROUTINE DELETE(NROWS.LIST)
      INTEGER NPATCH(9.4.4.3).FTOTAL,FGRD(3),GINVEN(2.6.4.5.3).
     *FPAT(3),NGRD(3),GRA(3.2).FWID(3,4).XTOTAL.XGRD(3).XWID(3.4).
     *GD(10.2).DINVEN(11.4,5.3),COMPL,GRD,THK.LEN.TYP.DDATE(3).
     *SS1.SS2.CODE.PD.P,TAIL(660),HEAD(660),LINKPR(1500).NAR,ROW.
     *LINKSR(1500).MASTER(1500.3).AHEAD.BEHIND.IHEAD.ITAIL.ITEM.SIZE.
     *NROWS,DLT(75).LSIZE(660).TRNSFR(3).PROW(26).NM.TROW.LN.
     *LENGTH(11).VENDV(8.4.4).CNT.LIST.CNTR.VCODE
      REAL QTY.ORIQTY.FRC(11.2),PRATE(3).JRATE(4.4)
      LOGICAL FILLED
      BYTE GRADE(11.4),THICK(11.4).WID(11.2)
      COMMON /MODEL7/ NPATCH.GINVEN,VENDV.DINVEN.JRATE,FTOTAL.FGRD.
     *FPAT.FWID.NGRD.GD.LN.GRA,XTOTAL.XGRD.XWID.CNT.CNTR.VCODE
      COMMON /MODEL8/ GRD.THK.LEN.TYP.DDATE.ORIQTY.SS1.SS2.CODE.QTY.
     *FRC.PRATE.PD.P.COMPL,NM,PROW.TROW.FILLED.GRADE.THICK.LENGTH.
     *WID
      COMMON /MODEL9/ TAIL,HEAD.LINKPR,MASTER.NAR.ROW.AHEAD.BEHIND.
     *IHEAD.ITAIL,ITEM,DLT.SIZE,LINKSR,TRNSFR,LSIZE
C
C    IF ALL THE ENTRIES OF A LIST ARE TO BE DELETED. THEN RESET HEAD AND
C    TAIL POINTER TO ZERO AND REDUCE LIST SIZE TO ZERO.
C
      IF(LSIZE(LIST).GT.NROWS)GO TO 10
      LSIZE(LIST)=0
      HEAD(LIST)=0
      TAIL(LIST)=0
      GO TO 15
C
C    ONLY A FEW ENTRIES OF THE CURRENT LIST ARE TO BE REMOVED
C
10    LSIZE(LIST)=LSIZE(LIST)-NROWS
      ROW=DLT(NROWS)
      ITAIL=LINKPR(ROW)
      LINKSR(ITAIL)=0
      TAIL(LIST)=ITAIL
C
C    PLACING ENTRIES IN THE LIST OF AVAILABLE SPACE
C
15    DO 20 I=1.NROWS
      LINKSR(DLT(I))=NAR
      LINKPR(DLT(I))=0
      NAR=DLT(I)
20    CONTINUE
      RETURN
      END
```

```
C
C     FEB. 21.1984
C
C     THIS SUBROUTINE IS A SUPPLEMENT TO SUB. INICNT.  IT ATTEMPTS TO FILL
C     THE NARROW WIDTH ORDERS BY SEARCHING DRY INVENTORY USING EACH OF
C     THE ALTERNATIVE VENEER SOURCES DEFINED IN SUB. INICNT.  IF AN
C     ALTERNATIVE IS NOT FOUND. THE NEXT ALTERNATIVE IS CHECKED UNTIL
C     EITHER THE ORDER IS FILLED OR THE VENEER DEVELOPMENT FILE HAS BEEN
C     EXHAUSTED.
C
C   AUTHOR:  ERIC SCHILD
C
      SUBROUTINE INIXBD
      INTEGER NPATCH(9.4.4.3),ETOTAL.EGRD(3),GINVEN(2.6,4.5.3).
     *EPAT(3),NGRD(3),GRA(3,2).EWID(3,4),XTOTAL.XGRD(3),XWID(3,4).
     *GD(10.2).DINVEN(11,4.5.3),COMPL.GRD,THK.LEN.TYP.DDATE(3).
     *SS1.SS2.CODE.PD.P,TAIL(660).HEAD(660).LINKPR(1500).NAR.ROW.
     *LINKSR(1500).MASTER(1500.3),AHEAD.BEHIND.IHEAD.ITAIL.ITEM.SIZE.
     *NROWS.DLT(75).LSIZE(660).TRNSFR(3).PROW(26).NM.TROW,LN,
     *LENGTH(11).CD.NT.NR.VENDV(8.4.4).CNT.L(3).LG,BD.CNTR.VCODE
      REAL QTY.F(4.3),ORIQTY,FRC(11.2),PRATE(3),JRATE(4.4)
      LOGICAL FILLED
      BYTE GRADE(11.4).THICK(11.4).WID(11.2)
      COMMON /MODEL7/ NPATCH.GINVEN,VENDV,DINVEN.JRATE.ETOTAL.EGRD,
     *EPAT.EWID.NGRD.GD.LN.GRA.XTOTAL.XGRD.XWID.CNT.CNTR.VCODE
      COMMON /MODEL8/ GRD,THK.LEN.TYP.DDATE,ORIQTY.SS1.SS2,CODE.QTY.
     *FRC.PRATE.PD.P.COMPL.NM.PROW.TROW.FILLED.GRADE.THICK.LENGTH.
     *WID
      COMMON /MODEL9/ TAIL.HEAD.LINKPR.MASTER.NAR.ROW,AHEAD.BEHIND.
     *IHEAD,ITAIL,ITEM.DLT.SIZE,LINKSR.TRNSFR.LSIZE
C
C   CONVERSION FACTORS USED TO CALCULATE THE VENEER NEEDS
C
      DATA L/5.1.4/
      DATA F/2..2.25,2.5.3.,1.,1.125,1.25,1.5,.67..75..83.1./
C
C   ESTABLISHING THE TOTAL NUMBER OF WIDTHS THAT CAN BE USED WITH THE
C   CURRENT GRADE
C
      BD=XWID(CNT.1)
      DO 35 CNTR=1.BD
C
C    VENEER WIDTH
C
      SS2=XWID(CNT.CNTR+1)
C
C    CHECKING VARIOUS LENGTHS FOR CURRENT GRADE AND WIDTH
C
      DO 20 LG=1.3
      LN=L(LG)
C
C    DETERMINING DRY INVENTORY CODE
C
      CODE=DINVEN(SS1.THK.LN.SS2)
      IF(CODE.EQ.0)GO TO 20
C
C    CHECKING LINKED STORAGE ARRAY
C
      IF(LSIZE(CODE).EQ.0) GO TO 20
C
```

```
      ORIQTY=ORIQTY*F(LEN,LG)
      QTY=ORIQTY
      CALL NFIND(DDATE(1),CODE,ORIQTY)
      NT=1
      NR=0
      IF(ORIQTY.GE.QTY)GO TO 15
      QTY=QTY-ORIQTY
      CD=THK*100+LN*10+SS2
      WRITE(7,10) DDATE(1),QTY,CODE,CD.NT.NR
10    FORMAT(T2,I6.F6.2.2I4,2I2)
15    ORIQTY=ORIQTY/F(LEN.LG)
      IF(ORIQTY.GT.0.) GO TO 20
      FILLED=.TRUE.
      RETURN
20    CONTINUE
35    CONTINUE
      RETURN
      END
```

```
C
C    FEB. 21,1984
C
C    THIS SUBPROGRAM IS USED TO ENUMERATE ALL OF THE POSSIBLE PROCESSING
C    ALTERNATIVES THAT ARE AVAILABLE IN UPGRADING THE ALTERVATIVE SOURCE
C    TO THE DESIRED GRADE.
C
C    AUTHOR:  ERIC SCHILD
C
      SUBROUTINE BCKPS1
       INTEGER NPATCH(9,4,4,3),FTOTAL,FGRD(3),GINVEN(2,6,4,5,3),
     *FPAT(3),NGRD(3),GRA(3,2),FWID(3,4),XTOTAL,XGRD(3),XWID(3,4),
     *GD(10,2),DINVEN(11,4,5,3),COMPL,GRD,THK,LEN,TYP,DDATE(3),
     *SS1,SS2,CODE,PD,P,TAIL(660),HEAD(660),LINKPR(1500),NAR,ROW,
     *LINKSR(1500),MASTER(1500,3),AHEAD,BEHIND,IHEAD,ITAIL,ITEM,SIZE,
     *NROWS,DLT(75),LSIZE(660),TRNSFR(3),PROW(26),NM,TROW,LN,
     *LENGTH(11),VENDV(8,4,4),CNT,WIDE,BD,CNTR,VCODE,DUM(16)
      REAL QTY,ORIQTY,FRC(11,2),PRATE(3),JRATE(4,4)
      LOGICAL FILLED
      BYTE GRADE(11,4),THICK(11,4),WID(11,2)
      COMMON /MODEL7/ NPATCH,GINVEN,VENDV,DINVEN,JRATE,FTOTAL,FGRD,
     *FPAT,FWID,NGRD,GD,LN,GRA,XTOTAL,XGRD,XWID,CNT,CNTR,VCODE
      COMMON /MODEL8/ GRD,THK,LEN,TYP,DDATE,ORIQTY,SS1,SS2,CODE,QTY,
     *FRC,PRATE,PD,P,COMPL,NM,PROW,TROW,FILLED,GRADE,THICK,LENGTH,
     *WID
      COMMON /MODEL9/ TAIL,HEAD,LINKPR,MASTER,NAR,ROW,AHEAD,BEHIND,
     *IHEAD,ITAIL,ITEM,DLT,SIZE,LINKSR,TRNSFR,LSIZE
C
C    READING IN THE ALTERNATIVE SOURCES FROM THE VENEER DEVELOPMENT FILE.
C
      VCODE=VENDV(GRD,THK,LEN)
      READ(4'VCODE) FTOTAL,FGRD(1),FPAT(1),NGRD(1),(GRA(1,M),M=1,2),
     *FGRD(2),FPAT(2),NGRD(2),(GRA(2,M),M=1,2),FGRD(3),FPAT(3),
     *NGRD(3),(GRA(3,M),M=1,2),((FWID(N,M),M=1,4),N=1,3),(DUM(M),
     *M=1,16)
C
C    IF THE ORDER INVOLVES AN A, B, M, OR CP GRADE VENEER, THEN CONSTRUCT
C    A PATCHER TASK.
C
      IF(.NOT.((GRD.EQ.3).OR.(GRD.EQ.4).OR.(GRD.EQ.7).OR.
     *(GRD.EQ.8))) GO TO 10
C
      CALL PATGRD
      RETURN
C
C    PATCHING TASK IS NOT DEFINITELY REQUIRED
C
C    ENUMERATING THE ALTERNATIVE SOURCES
C
10    BD=FTOTAL
      DO 30 CNT=1,BD
      SS1=FGRD(CNT)
C
C    CHECKING THE WIDTHS FOR EACH ALTERNATIVE SOURCE
C
      WIDE=FWID(CNT,1)
C
      DO 25 CNTR=1,WIDE
      SS2=FWID(CNT,CNTR+1)
C
```

```
C     IF ALTERNATIVE SOURCE REQUIRES PATCHING, THEN CREATE PATCHER TASK
C     IF THE SOURCE IS AVAILABLE IN INVENTORY.
C
        IF(FPAT(CNT).EQ.0) GO TO 20
C
C     CHECKING TO SEE IF THE ALTERNATIVE ALSO REQUIRES JOINTING
C
        IF(SS2.GT.1) GO TO 15
C
C     ONLY PATCHING IS REQUIRED
C
        CALL PATCH
        IF(FILLED)RETURN
        GO TO 25
C
C     BOTH PATCHING AND JOINTING ARE REQUIRED
C
15      CALL PATJOT
        IF(FILLED)RETURN
        GO TO 25
C
C     NO PATCHING IS REQUIRED BUT DOES ALTERNATIVE REQUIRE JOINTING?
C
20      IF(SS2.GT.1)GO TO 22
C
C     NO JOINTING IS REQUIRED
C
        CALL DGRADE
        IF(FILLED)RETURN
        GO TO 25
C
22      CALL JOINT
        IF(FILLED)RETURN
C
25      CONTINUE
30      CONTINUE
        SS2=1
        RETURN
        END
```

```
C
C    FEB. 21,1984
C
C    THIS SUBPROGRAM GENERATES A PATCH TASK FOR THE CURRENT ORDER AND
C    DETERMINES WHETHER THE INTERMEDIATE GRADE CAN BE OBTAINED DIRECTLY
C    FROM DRY INVENTORY OR BY JOINTING A NARROW VENEER OF THE SAME
C    GRADE.
C
C    AUTHOR:  ERIC SCHILD
C
      SUBROUTINE PATGRD
      INTEGER NPATCH(9.4.4.3),FTOTAL.FGRD(3).GINVEN(2.6.4.5.3),
     *FPAT(3),NGRD(3).GRA(3.2).FWID(3,4),XTOTAL.XGRD(3),XWID(3,4),
     *GD(10.2).DINVEN(11.4.5.3).COMPL.GRD.THK.LEN.TYP.DDATE(3).
     *SS1.SS2.CODE,PD.P.TAIL(660),HEAD(660).LINKPR(1500).NAR.ROW,
     *LINKSR(1500).MASTER(1500.3).AHEAD.BEHIND.IHEAD.ITAIL.ITEM.SIZE,
     *NROWS.DLT(75).LSIZE(660),TRNSFR(3).PROW(26).NM.TROW,LN,
     *LENGTH(11),VENDV(8.4,4).CNT.PCODE.PROCTM.CNTR.VCODE.PROCT1.
     *PROCT2.NMBR.NG,GR(2),NT.NR.CD,C
      REAL QTY.TIMA.TIMC.ORIQTY.FRC(11.2).PRATE(3).JRATE(4.4).
     *TQTY.SQTY,PROP,TQ
      LOGICAL FILLED
      BYTE GRADE(11,4),THICK(11,4),WID(11.2)
      COMMON /MODEL7/ NPATCH.GINVEN,VENDV,DINVEN.JRATE.FTOTAL.FGRD,
     *FPAT.FWID.NGRD,GD.LN.GRA.XTOTAL.XGRD.XWID.CNT.CNTR.VCODE
      COMMON /MODEL8/ GRD,THK,LEN.TYP.DDATE,ORIQTY.SS1.SS2.CODE,QTY.
     *FRC.PRATE.PD.P.COMPL.NM.PROW.TROW.FILLED.GRADE.THICK,LENGTH.
     *WID
      COMMON /MODEL9/ TAIL.HEAD,LINKPR.MASTER.NAR.ROW.AHEAD.BEHIND.
     *IHEAD.ITAIL.ITEM.DLT.SIZE,LINKSR.TRNSFR,LSIZE
C
      SS1=FGRD(1)
      CNT=1
      NG=NGRD(1)
      GR(1)=GRA(1,1)
      GR(2)=GRA(1.2)
      DO 50 I=1.3
      SS2=I
C
C   DETERMINING DRY INVENTORY CODE AND PATCHER DEVELOPMENT CODE
C
      CODE=DINVEN(SS1.THK,LEN.SS2)
      PCODE=NPATCH(SS1.THK.LEN.SS2)
      IF(PCODE.EQ.0)GO TO 50
C
C    READING PATCHER DEVELOPMENT BREAKDOWN AND PATCHER RATE
C
      READ (2'PCODE) (FRC(GD(M,1).1).M=1.8),PRATE(2)
C
      PROP=0.
      DO 5 J=1.NG
      PROP=PROP+FRC(GR(J),1)
5     CONTINUE
C
C    ERROR CHECK-PATCHER BREAKDOWN FRACTION IS ZERO
C
      IF(PROP.EQ.0.) GO TO 50
C
C    DETERMINING REQUIRED QUANTITY THAT WILL BE PATCHED TO FILL THE
C    CURRENT ORDER
```

```
C
      TQTY=ORIQTY/PROP
      NT=1
      NR=0
      DO 15 J=1,NG
      CD=DINVEN(GR(J),THK,LEN,1)
      TQ=TQTY*FRC(GR(J),1)
      C=THK*100+LEN*10+1
      WRITE(7,10) DDATE(1),TQ,CD,C,NT,NR
10    FORMAT(T2,I6,F6.2,2I4,2I2)
15    CONTINUE
C
C    IF THE TOTAL QUANTITY EXCEEDS 10 UNITS, THEN BREAK THE CURRENT
C    PATCHER TASKS INTO SMALLER SUBTASKS OF 10 UNITS EACH
C
      TIMA=TQTY/10.
      TIMB=IFIX(TIMA)
      TIMC=TIMA-FLOAT(TIMB)
      NMBR=NINT(TIMA)
      DO 45 J=1,NMBR
      ORIQTY=10.
      IF(J.LT.NMBR)GO TO 20
      ORIQTY=TIMC*10.
C
C    COMPUTING PROCESSING TIME ON PATCHER OPERATION
C
20    PROCTM=NINT(ORIQTY/PRATE(2))
C
C    START TIME ON PATCHER OPERATION
C
      DDATE(2)=DDATE(1)-PROCTM
C
C    IF NARROW WIDTH VENEER IS TO BE USED TO PROCURE THE CURRENT ORDER.
C    THEN IN ADDITION TO A PATCHER TASK , A JOINTER TASK IS ALSO NEEDED.
C
      IF(SS2.GT.1) GO TO 30
C
C   CHECKING DRY INVENTORY FOR THE ALTERNATIVE SOURCE
C
      QTY=ORIQTY
      IF(LSIZE(CODE).EQ.0) GO TO 25
      CALL NFIND(DDATE(2),CODE,ORIQTY)
      P=2
      CALL QTYCK1
C
C    IF THE CURRENT ORDER IS FILLED WITH DRY INVENTORY, THEN
C    CHECK NEXT SUBTASK.
C
      IF(FILLED)GO TO 40
C
C    OTHERWISE DEVELOP THE NEEDED DRY INVENTORY FROM GREEN INVENTORY
C    WITH A DRYER TASK.
C
25    GRD=SS1
C
C    COMPUTING PROCESSING TIME ON PATCHER OPERATION
C
      PROCTM=NINT(ORIQTY/PRATE(2))
C
C    START TIME ON PATCHER OPERATION
```

```
C
      DDATE(1)=DDATE(1)-PROCTM
      TQTY=ORIQTY
      CALL DRYER1
C
C   RECALCULATING ORIGINAL DUE DATE FOR ORDER AT SPREADER OPERATION
C
      DDATE(1)=DDATE(1)+PROCTM
C
C   CALLING SLAVE ROUTINE TO SCHEDULE CURRENT SUBTASK ON ONE OF THE
C   MACHINES OF THE PATCHER OPERATION
C
      CALL TRNSMT(1,2,PROCTM,GRD,THK,LEN,SS2,CODE,DDATE(1),
     *TROW,PRATE(2),NM,PROW)
      PD=1
      COMPL=DDATE(1)
      QTY=TQTY
      NGRD(1)=NG
      GRA(1,1)=GR(1)
      GRA(1,2)=GR(2)
      CALL INVDEV
      GO TO 40
C
C  CREATING A PATCHER AND A JOINTER TASK
C
30    PRATE(3)=JRATE(THK,LEN)
C
C   CHECKING DRY INVENTORY FOR THE ALTERNATIVE SOURCE
C
C   COMPUTING PROCESSING TIME ON JOINTER OPERATION
C
      PROCTM=NINT(ORIQTY/PRATE(3))
C
C   START TIME ON JOINTER OPERATION
C
      DDATE(3)=DDATE(2)-PROCTM
      QTY=ORIQTY
      IF(LSIZE(CODE).EQ.0)GO TO 35
      CALL NEIND(DDATE(3),CODE,ORIQTY)
      CALL QTYCK2
C
C   IF FILLED, CHECK THE NEXT SUBTASK
C
      IF(FILLED)GO TO 40
C
C   OTHERWISE OBTAIN THE DESIRED DRY VENEER FROM GREEN INVENTORY BY
C   CREATING A DRYER TASK
C
35    GRD=SS1
C
C   COMPUTING PROCESSING TIME ON PATCHER OPERATION
C
      PROCT1=NINT(ORIQTY/PRATE(2))
C
C   START TIME ON PATCHER OPERATION
C
      DDATE(2)=DDATE(1)-PROCT1
C
C   COMPUTING PROCESSING TIME ON JOINTER OPERATION
C
```

```
      PROCT2=NINT(ORIQTY/PRATE(3))
C
C    START TIME ON JOINTER OPERATION
C
      DDATE(3)=DDATE(2)-PROCT2
      DDATE(1)=DDATE(3)
      IQTY=ORIQTY
      CALL DRYER1
C
C    CALLING THE SLAVE ROUTINE TO SCHEDULE THE SUBTASK ON ONE OF THE
C    MACHINES OF THE JOINTER OPERATION
C
      CALL TRNSMT(1,3,PROCT2,GRD,THK,LEN,SS2,CODE,DDATE(2),
     *TROW,PRATE(3),NM,PROW)
      PROW(1)=TROW
      NM=1
      DDATE(1)=DDATE(2)+PROCT1
C
C    CALLING THE SLAVE ROUTINE TO SCHEDULE THE SUBTASK ON ONE OF THE
C    MACHINES OF THE PATCHER OPERATION
C
      CALL TRNSMT(1,2,PROCT1,GRD,THK,LEN,SS2,CODE,DDATE(1),
     *TROW,PRATE(2),NM,PROW)
      PD=1
      COMPL=DDATE(1)
      QTY=IQTY
      SS2=1
      NGRD(1)=NG
      GRA(1,1)=GR(1)
      GRA(1,2)=GR(2)
      CALL INVDEV
40    FILLED=.FALSE.
45    CONTINUE
      FILLED=.TRUE.
      RETURN
50    CONTINUE
      WRITE(5,55)
55    FORMAT(5X,'TASK CANNOT BE FILLED WITH AVAILABLE SOURCES')
      STOP
      END
```

```
C
C    FEB. 21.1984
C
C    THIS SUBPROGRAM SETS UP BOTH A PATCHER AS WELL AS A JOINTER TASK TO
C    DEVELOP THE ALTERNATIVE SOURCE INTO THE DESIRED GRADE IF THE REQUIR-
C    ED SOURCE IS FOUND IN DRY INVENTORY.
C
C    AUTHOR:  ERIC SCHILD
C
      SUBROUTINE PATJOT
      INTEGER NPATCH(9,4,4.3),FTOTAL,FGRD(3).GINVEN(2,6,4.5,3).
     *FPAT(3).NGRD(3).GRA(3.2).FWID(3,4).XTOTAL.XGRD(3).XWID(3.4).
     *GD(10.2).DINVEN(11,4.5.3).COMPL,GRD.THK.LEN.TYP.DDATE(3).
     *SS1.SS2.CODE.PD.P.TAIL(660),HEAD(660),LINKPR(1500).NAR.ROW.
     *LINKSR(1500),MASTER(1500,3).AHEAD.BEHIND.IHEAD.ITAIL.ITEM.SIZE.
     *N[illegible]MS.DLT(75),LSIZE(660),TRNSFR(3).PROW(26),NM.TROW,LN,
     *LENGTH(11).VENDV(8.4,4).CNT.PCODE.PROCTM,NMBR.CNTR.VCODE.
     *NT.NR.CD
      REAL QTY.PROP.TIMA,TIMC.ORIQTY,FRC(11.2).PRATE(3).JRATE(4.4).
     *TQTY,SQTY.TQ.AMT
      LOGICAL FILLED
      BYTE GRADE(11.4),THICK(11,4),WID(11.2)
      COMMON /MODEL7/ NPATCH.GINVEN,VENDV.DINVEN,JRATE,FTOTAL,FGRD.
     *FPAT.FWID,NGRD.GD.LN.GRA.XTOTAL.XGRD.XWID.CNT.CNTR,VCODE
      COMMON /MODEL8/ GRD.THK,LEN.TYP.DDATE,ORIQTY.SS1.SS2.CODE.QTY.
     *FRC.PRATE.PD.P.COMPL.NM.PROW.TROW.FILLED.GRADE.THICK.LENGTH.
     *WID
      COMMON /MODEL9/ TAIL.HEAD.LINKPR.MASTER.NAR.ROW.AHEAD.BEHIND.
     *IHEAD.ITAIL.ITEM.DLT.SIZE.LINKSR.TRNSFR.LSIZE
C
C    SETTING UP PATCHER DEVELOPMENT CODE
C
      PCODE=NPATCH(SS1,THK.LEN.SS2)
      IF(PCODE.EQ.0)RETURN
      CODE=DINVEN(SS1.THK.LEN.SS2)
C
C    READING PATCHER DEVELOPMENT BREAKDOWN AND PATCHER RATE
C
      READ (2'PCODE) (FRC(GD(M.1).1).M=1.8).PRATE(2)
C
      PROP=0.
      DO 5 I=1.NGRD(CNT)
      PROP=PROP+FRC(GRA(CNT.I).1)
5     CONTINUE
C
C    ERROR CHECK-IF BREAKDOWN FRACTION IS ZERO. THEN THE PATCHER CODING
C    SCHEME IS IN ERROR
C
      IF(PROP.GT.0.)GO TO 10
      WRITE(5.8)
8     FORMAT(I2.'ZERO DIVIDE ERROR-PATCHER DEVELOPMENT FILE'/
     *I2.'SOURCE-SUB. PATJOT')
      RETURN
C
C    DETERMINING TOTAL QUANTITY REQUIRED AT THE PATCHER OPERATION TO
C    FILL THE CURRENT ORDER
C
10    TQ=ORIQTY
      TQTY=ORIQTY/PROP
C
```

```
C    IF THE TOTAL QUANTITY EXCEEDS 10 UNITS, THEN BREAK THE TASK UP INTO
C    SMALLER SUBTASKS OF 10 UNITS EACH
C
      TIMA=TQTY/10.
      TIMB=IFIX(TIMA)
      TIMC=TIMA-FLOAT(TIMB)
      NMBR=NINT(TIMA)
      DO 20 I=1,NMBR
      ORIQTY=10.
      IF(I.LT.NMBR)GO TO 15
      ORIQTY=TIMC*10.
C
C    CHECKING DRY INVENTORY FOR THE ALTERNATIVE SOURCE
C
15    IF(LSIZE(CODE).EQ.0)GO TO 25
C
C    COMPUTING PROCESSING TIME ON PATCHER OPERATION
C
      PROCTM=NINT(ORIQTY/PRATE(2))
C
C    START TIME ON PATCHER OPERATION
C
      DDATE(2)=DDATE(1)-PROCTM
      PRATE(3)=JRATE(THK,LEN)
C
C    COMPUTING PROCESSING TIME ON JOINTER OPERATION
C
      PROCTM=NINT(ORIQTY/PRATE(3))
C
C    START TIME ON JOINTER OPERATION
C
      DDATE(3)=DDATE(2)-PROCTM
      SQTY=ORIQTY
      QTY=ORIQTY
      CALL NFIND(DDATE(3),CODE,ORIQTY)
      CALL QTYCK2
      TQTY=TQTY-(SQTY-ORIQTY)
20    CONTINUE
C
C    RECALCULATE CURRENT ORDER QUANTITY
C
25    ORIQTY=TQTY*PROP
      IF(ORIQTY.GE.TQ)GO TO 40
      NT=1
      NR=0
      TQ=TQ-ORIQTY
      TQ=TQ/PROP
      DO 35 I=1,NGRD(CNT)
      CODE=DINVEN(GRA(CNT,I),THK,LEN,1)
      AMT=TQ*FRC(GRA(CNT,I),1)
      CD=THK*100+LEN*10+1
      WRITE(7,30) DDATE(1),AMT,CODE,CD,NT,NR
30    FORMAT(I2,I6,F6.2,2I4,2I2)
35    CONTINUE
40    FILLED=.FALSE.
C
C    IF CURRENT ORDER HAS BEEN FILLED, SET FILLED TO TRUE
C
      IF(ORIQTY.GT.0)RETURN
      FILLED=.TRUE.
```

```
RETURN
END
```

```
C
C   FEB. 21,1984
C
C   IF THE ALTERNATIVE SOURCE REQUIRES IT, THIS SUBPROGRAM IS USED TO
C   EVALUATE THE PATCHING OPTION.  THE SUBROUTINE FIRST COMPUTES THE
C   TOTAL QUANTITY NEEDED TO FILL THE CURRENT ORDER AND THEN CHECKS DRY
C   INVENTORY FOR THE ALTERNATIVE SOURCE.
C
C   AUTHOR:  ERIC SCHILD
C
      SUBROUTINE PATCH
      INTEGER NPATCH(9,4,4,3),FTOTAL,FGRD(3),GINVEN(2,6,4,5,3),
     *FPAT(3),NGRD(3),GRA(3,2),FWID(3,4),XTOTAL,XGRD(3),XWID(3,4),
     *GD(10,2),DINVEN(11,4,5,3),COMPL,GRD,THK,LEN,TYP,DDATE(3),
     *SS1,SS2,CODE,PD,P,TAIL(660),HEAD(660),LINKPR(1500),NAR,ROW,
     *LINKSR(1500),MASTER(1500,3),AHEAD,BEHIND,IHEAD,ITAIL,ITEM,SIZE,
     *NROWS,DLT(75),LSIZE(660),TRNSFR(3),PROW(26),NM,TROW,LN,
     *LENGTH(11),VENDV(8,4,4),CNT,PCODE,PROCTM,NMBR,NT,NR,CNTR,VCODE,
     *CD
      REAL QTY,TIMA,TIMC,TQTY,ORIQTY,FRC(11,2),PRATE(3),JRATE(4,4),
     *SQTY,TQ,AMT,PROP
      LOGICAL FILLED
      BYTE GRADE(11,4),THICK(11,4),WID(11,2)
      COMMON /MODEL7/ NPATCH,GINVEN,VENDV,DINVEN,JRATE,FTOTAL,FGRD,
     *FPAT,FWID,NGRD,GD,LN,GRA,XTOTAL,XGRD,XWID,CNT,CNTR,VCODE
      COMMON /MODEL8/ GRD,THK,LEN,TYP,DDATE,ORIQTY,SS1,SS2,CODE,QTY,
     *FRC,PRATE,PD,P,COMPL,NM,PROW,TROW,FILLED,GRADE,THICK,LENGTH,
     *WID
      COMMON /MODEL9/ TAIL,HEAD,LINKPR,MASTER,NAR,ROW,AHEAD,BEHIND,
     *IHEAD,ITAIL,ITEM,DLT,SIZE,LINKSR,TRNSFR,LSIZE
C
C   DETERMINING DRY INVENTORY AND PATCHER DEVELOPMENT CODE
C
      SS2=1
      CODE=DINVEN(SS1,THK,LEN,SS2)
      PCODE=NPATCH(SS1,THK,LEN,SS2)
      IF(PCODE.EQ.0)RETURN
C
C   READING PATCHER DEVELOPMENT BREAKDOWN AND PROCESSING RATE
C
      READ(2'PCODE) (FRC(GD(M,1),1),M=1,8),PRATE(2)
C
      PROP=0.
      DO 5 I=1,NGRD(CNT)
      PROP=PROP+FRC(GRA(CNT,I),1)
5     CONTINUE
C
C  ERROR CHECK-IF PATCHER BREAKDOWN IS ZERO, RETURN
C
      IF(PROP.GT.0.)GO TO 15
      WRITE(5,10)
10    FORMAT(T2,'ZERO DIVIDE ERROR-PATCHER DEVELOPMENT FILE'/
     *T2,'SOURCE-SUB. PATCH')
      RETURN
C
C   IF TOTAL QUANTITY EXCEEDS 10 UNITS, BREAK THE TASK UP INTO SMALLER
C   SUBTASKS OF 10 UNITS EACH
C
15    TQ=ORIQTY
      TQTY=ORIQTY/PROP
```

```
      TIMA=TQTY/10.
      TIMB=IFIX(TIMA)
      TIMC=TIMA-FLOAT(TIMB)
      NMBR=NINT(TIMA)
      DO 25 I=1,NMBR
      ORIQTY=10.
      IF(I.LT.NMBR)GO TO 20
      ORIQTY=TIMC*10.
C
C   CHECKING DRY INVENTORY FOR ALTERNATIVE SOURCE
C
20    IF(LSIZE(CODE).EQ.0)GO TO 30
C
C   COMPUTING PROCESSING TIME
C
      PROCTM=NINT(ORIQTY/PRATE(2))
      P=2
C
C   START TIME FOR TASK ON PATCHER OPERATION
C
      DDATE(2)=DDATE(1)-PROCTM
      SQTY=ORIQTY
      QTY=ORIQTY
      CALL NEIND(DDATE(2),CODE,ORIQTY)
      CALL QTYCK1
      TQTY=TQTY-(SQTY-ORIQTY)
25    CONTINUE
C
C   RECALCULATING ORDER QUANTITY
C
30    ORIQTY=TQTY*PROP
      IF(ORIQTY.GE.TQ)GO TO 50
      NT=1
      NR=0
      TQ=TQ-ORIQTY
      TQ=TQ/PROP
      DO 40 I=1,NGRD(CNT)
      CODE=DINVEN(GRA(CNT,I),THK,LEN,SS2)
      AMT=TQ*FRC(GRA(CNT,I),1)
      CD=THK*100+LEN*10+SS2
      WRITE(7,35) DDATE(1),AMT,CODE,CD,NT,NR
35    FORMAT(I2,I6,F6.2,2I4,2I2)
40    CONTINUE
50    FILLED=.FALSE.
C
C   IF CURRENT ORDER HAS BEEN FILLED, SET FILLED TO TRUE
C
      IF(ORIQTY.GT.0)RETURN
      FILLED=.TRUE.
      RETURN
      END
```

```
C
C    FEB. 21.1984
C
C    THIS SUBPROGRAM IS USED TO DEVELOP THE ALTERNATIVE SOURCE INTO THE
C    DESIRED GRADE USING THE JOINTER OPERATION IF THE SOURCE IS AVAIL-
C    ABLE IN DRY INVENTORY.
C
C    AUTHOR:  ERIC SCHILD
C
      SUBROUTINE JOINT
      INTEGER NPATCH(9,4,4,3),FTOTAL,FGRD(3),GINVEN(2,6,4,5,3),
     *FPAT(3),NGRD(3),GRA(3,2),FWID(3,4),XTOTAL,XGRD(3),XWID(3,4),
     *GD(10,2),DINVEN(11,4,5,3),COMPL,GRD,THK,LEN,TYP,DDATE(3),
     *SS1,SS2,CODE,PD,P,TAIL(660),HEAD(660),LINKPR(1500),NAR,ROW,
     *LINKSR(1500),MASTER(1500,3),AHEAD,BEHIND,IHEAD,ITAIL,ITEM,SIZE,
     *NROWS,DLT(75),LSIZE(660),TRNSFR(3),PROW(26),NM,TROW,LN,
     *LENGTH(11),VENDV(8,4,4),CNT,PROCTM,NMBR,NT,NR,CD,CNTR,VCODE
      REAL QTY,TQ,TQTY,SQTY,TIMC,ORIQTY,FRC(11,2),PRATE(3),JRATE(4,4),
     *TIMA
      LOGICAL FILLED
      BYTE GRADE(11,4),THICK(11,4),WID(11,2)
      COMMON /MODEL7/ NPATCH,GINVEN,VENDV,DINVEN,JRATE,FTOTAL,FGRD,
     *FPAT,FWID,NGRD,GD,LN,GRA,XTOTAL,XGRD,XWID,CNT,CNTR,VCODE
      COMMON /MODEL8/ GRD,THK,LEN,TYP,DDATE,ORIQTY,SS1,SS2,CODE,QTY,
     *FRC,PRATE,PD,P,COMPL,NM,PROW,TROW,FILLED,GRADE,THICK,LENGTH,
     *WID
      COMMON /MODEL9/ TAIL,HEAD,LINKPR,MASTER,NAR,ROW,AHEAD,BEHIND,
     *IHEAD,ITAIL,ITEM,DLT,SIZE,LINKSR,TRNSFR,LSIZE
C
C    DETERMINING DRY INVENTORY CODE
C
      CODE=DINVEN(SS1,THK,LEN,SS2)
C
C    JOINTER PRODUCTION RATE
C
      PRATE(3)=JRATE(THK,LEN)
C
C    IF ORDER QUANTITY EXCEEDS 10 UNITS, BREAK THE TASK UP INTO SMALLER
C    SUBTASKS OF 10 UNITS EACH
C
      TQ=ORIQTY
      TQTY=ORIQTY
      TIMA=TQTY/10.
      TIMB=IFIX(TIMA)
      TIMC=TIMA-FLOAT(TIMB)
      NMBR=NINT(TIMA)
      DO 15 I=1,NMBR
      ORIQTY=10.
      IF(I.LT.NMBR)GO TO 10
      ORIQTY=TIMC*10.
C
C    CHECKING DRY INVENTORY FOR ALTERNATIVE SOURCE
C
10    IF(LSIZE(CODE).EQ.0)GO TO 20
C
C    COMPUTING PROCESSING TIME
C
      PROCTM=NINT(ORIQTY/PRATE(3))
      P=3
C
```

```
C   START TIME OF JOINTER TASK
C
      DDATE(3)=DDATE(1)-PROCTM
      SQTY=ORIQTY
      QTY=ORIQTY
      CALL NFIND(DDATE(3),CODE,ORIQTY)
      CALL QTYCK1
      TQTY=TQTY-(SQTY-ORIQTY)
15    CONTINUE
20    ORIQTY=TQTY
C
      IF(ORIQTY.GE.TQ)GO TO 40
      NT=1
      NR=0
      TQ=TQ-ORIQTY
      CODE=DINVEN(GRD,THK,LEN,1)
      CD=THK*100+LEN*10+1
      WRITE(7,30) DDATE(1),TQ,CODE,CD,NT,NR
30    FORMAT(T2,I6,F6.2,2I4,2I2)
40    FILLED=.FALSE.
C
C   IF ORDER WAS FILLED, SET FILLED TO TRUE
C
      IF(ORIQTY.GT.0)RETURN
      FILLED=.TRUE.
      RETURN
      END
```

```
C
C    FEB. 21.1984
C
C    THIS SUBROUTINE IS SIMILAR TO SUB INICNT IN THAT IT IS USED TO
C    CHECK DRY INVENTORY FOR THE DESIRED GRADE.  THOUGH. INSTEAD OF
C    SEARCHING FOR THE DESIRED GRADE. THIS ROUTINE CHECKS DRY INVENTORY
C    FOR THE GRADE SPECIFIED BY THE VENEER DEVELOPMENT FILE.
C
C    AUTHOR:  ERIC SCHILD
C
      SUBROUTINE DGRADE
      INTEGER NPATCH(9.4.4.3),FTOTAL,FGRD(3).GINVEN(2,6.4.5.3),
     *FPAT(3).NGRD(3),GRA(3,2),FWID(3,4).XTOTAL,XGRD(3),XWID(3,4).
     *GD(10.2),DINVEN(11,4.5.3).COMPL.GRD,THK,LEN.TYP,DDATE(3).
     *SS1.SS2.CODE.PD.P.TAIL(660).HEAD(660),LINKPR(1500),NAR.ROW,
     *LINKSR(1500).MASTER(1500.3).AHEAD.BEHIND.IHEAD.ITAIL,ITEM.SIZE.
     *NROWS.DLT(75).LSIZE(660).TRNSFR(3).PROW(26).NM.TROW.LN,
     *LENGTH(11).VENDV(8.4.4),CNT.NT.NR.CD.CNTR.VCODE
      REAL QTY.ORIQTY.FRC(11,2).PRATE(3).JRATE(4.4)
      LOGICAL FILLED
      BYTE GRADE(11,4),THICK(11.4).WID(11.2)
      COMMON /MODEL7/ NPATCH,GINVEN.VENDV,DINVEN,JRATE,FTOTAL.FGRD,
     *FPAT.FWID,NGRD.GD.LN.GRA,XTOTAL.XGRD.XWID.CNT.CNTR.VCODE
      COMMON /MODEL8/ GRD.THK.LEN.TYP.DDATE.ORIQTY,SS1.SS2.CODE.QTY,
     *FRC.PRATE.PD.P.COMPL,NM.PROW.TROW.FILLED.GRADE.THICK.LENGTH.
     *WID
      COMMON /MODEL9/ TAIL,HEAD,LINKPR.MASTER.NAR.ROW.AHEAD.BEHIND.
     *IHEAD,ITAIL,ITEM.DLT,SIZE.LINKSR.TRNSFR,LSIZE
C
C    DETERMINING DRY INVENTORY CODE
C
      CODE=DINVEN(SS1,THK.LEN.SS2)
C
C    CHECKING DRY INVENTORY FOR ALTERNATIVE SOURCE
C
      IF(LSIZE(CODE).EQ.0)RETURN
      QTY=ORIQTY
      CALL NFIND(DDATE(1),CODE,ORIQTY)
      IF(ORIQTY.GE.QTY)GO TO 15
      NT=1
      NR=0
      QTY=QTY-ORIQTY
      CD=THK*100+LEN*10+SS2
      WRITE(7.10) DDATE(1),QTY.CODE,CD.NT.NR
10    FORMAT(T2.I6,F6.2,2I4.2I2)
C
15    IF(ORIQTY.GT.0.)RETURN
      FILLED=.TRUE.
      RETURN
      END
```

```
C
C     FEB. 21.1984
C
C     THIS SUBROUTINE WORKS IN CONJUNCTION WITH SUB. PATCH, JOINT, AND
C     PATGRD TO ACTUALLY CREATE THE RESPECTIVE TASKS.  IT INCORPORATES
C     THE SEARCH SUBPROGRAM. NFIND, TO CHECK INVENTORY UNTIL EITHER
C     THE ORDER IS FILLED OR THE DESIRED GRADE HAS BEEN DEPLETED FROM
C     INVENTORY.
C
C     AUTHOR:  ERIC SCHILD
C
        SUBROUTINE QTYCK1
        INTEGER NPATCH(9,4.4.3),FTOTAL.FGRD(3).GINVEN(2,6.4.5.3).
       *FPAT(3).NGRD(3),GRA(3,2),FWID(3,4).XTOTAL.XGRD(3),XWID(3,4),
       *GD(10.2).DINVEN(11,4,5.3).COMPL,GRD.THK.LEN.TYP.DDATE(3).
       *SS1,SS2.CODE.PD,P,TAIL(660).HEAD(660).LINKPR(1500).NAR.ROW,
       *LINKSR(1500),MASTER(1500.3).AHEAD.BEHIND.IHEAD.ITAIL.ITEM.SIZE.
       *NROWS.DLT(75),LSIZE(660),TRNSFR(3).PROW(26).NM.TROW.LN,
       *LENGTH(11).VENDV(8.4.4).CNT.PROCTM,CNTR.VCODE
        REAL QTY.ORIQTY.FRC(11.2).PRATE(3).JRATE(4.4)
        LOGICAL FILLED
        BYTE GRADE(11,4).THICK(11.4).WID(11.2)
        COMMON /MODEL7/ NPATCH,GINVEN,VENDV,DINVEN,JRATE,FTOTAL,FGRD.
       *FPAT,FWID.NGRD,GD.LN,GRA.XTOTAL.XGRD.XWID.CNT.CNTR,VCODE
        COMMON /MODEL8/ GRD.THK,LEN.TYP.DDATE.ORIQTY,SS1.SS2,CODE,QTY.
       *FRC.PRATE.PD.P.COMPL.NM.PROW.TROW.FILLED.GRADE.THICK.LENGTH.
       *WID
        COMMON /MODEL9/ TAIL,HEAD.LINKPR,MASTER.NAR.ROW.AHEAD.BEHIND.
       *IHEAD,ITAIL,ITEM.DLT,SIZE,LINKSR,TRNSFR,LSIZE
C
C     IF THE DESIRED GRADE WAS NOT FOUND IN INVENTORY. RETURN
C
5       IF(ORIQTY.EQ.QTY) GO TO 25
C
C     IF QUANTITY WAS NOT COMPLETELY FILLED. CONTINUE
C
        IF(ORIQTY.GT.0.) GO TO 15
C
C     ORDER WAS COMPLETELY FILLED
C
C     COMPUTING PROCESSING TIME FOR THE PTH OPERATION
C
        PROCTM=NINT(QTY/PRATE(P))
C
C     COMPLETION TIME FOR PTH OPERATION
C
        DDATE(P)=DDATE(P)+PROCTM
C
C     CALLING SLAVE ROUTINE TO SCHEDULE TASK ON ONE OF THE MACHINES OF
C     THE PTH OPERATION
C
        CALL TRNSMT(1,P,PROCTM.SS1.THK.LEN.SS2,CODE.DDATE(P).
       *TROW,PRATE(P),NM.PROW)
        IF(P.NE.2)GO TO 10
        PD=1
        COMPL=DDATE(P)
        CALL INVDEV
10      FILLED=.TRUE.
        RETURN
C
```

```
C     ORDER WAS NOT COMPLETELY FILLED
C
C     COMPUTING PROCESSING TIME FOR PTH OPERATION
C
15       QTY=QTY-ORIQTY
         PROCTM=NINT(QTY/PRATE(P))
C
C     COMPLETION TIME ON THE PTH OPERATION
C
         DDATE(P)=DDATE(P)+PROCTM
C
C     CALLING SLAVE ROUTINE TO SCHEDULE TASK ON ONE OF THE MACHINES OF
C     THE PTH OPERATION
C
         CALL TRNSMT(1,P,PROCTM,SS1,THK,LEN,SS2,CODE,DDATE(P),
        *TROW,PRATE(P),NM,PROW)
         IF(P.NE.2)GO TO 20
         PD=1
         COMPL=DDATE(P)
         CALL INVDEV
20       QTY=ORIQTY
C
C     CHECKING INVENTORY AGAIN WITH NEW STARTING TIME AND UPDATED ORDER
C     QUANTITY
C
C     IF DESIRED INVENTORY HAS BEEN EXHAUSTED, RETURN
C
         IF(LSIZE(CODE).EQ.0) RETURN
         CALL NFIND(DDATE(P),CODE,ORIQTY)
         GO TO 5
25       CONTINUE
         RETURN
         END
```

```
C
C  JUNE 4. 1984
C
C  THIS SUBROUTINE IS IDENTICAL IN NATURE TO SUBROUTINE QTYCK1 EXCEPT IN
C  THIS SUBPROGRAM TWO TASKS ARE CREATED IN A SINGLE PASS INSTEAD OF ONE
C  AS WAS THE CASE WITH QTYCK1.
C
C  AUTHOR:  ERIC SCHILD
C
      SUBROUTINE QTYCK2
      INTEGER NPATCH(9,4,4,3),FTOTAL,FGRD(3),GINVEN(2,6,4,5,3),
     *FPAT(3),NGRD(3),GRA(3,2),FWID(3,4),XTOTAL,XGRD(3),XWID(3,4),
     *GD(10,2),DINVEN(11,4,5,3),COMPL,GRD,THK,LEN,TYP,DDATE(3),
     *SS1,SS2,CODE,PD,P,TAIL(660),HEAD(660),LINKPR(1500),NAR,ROW,
     *LINKSR(1500),MASTER(1500,3),AHEAD,BEHIND,IHEAD,ITAIL,ITEM,SIZE,
     *NROWS,DLT(75),LSIZE(660),TRNSFR(3),PROW(26),NM,TROW,LN,
     *LENGTH(11),VENDV(8,4,4),CNT,PROCTM,CNTR,VCODE
      REAL QTY,ORIQTY,FRC(11,2),PRATE(3),JRATE(4,4)
      LOGICAL FILLED
      BYTE GRADE(11,4),THICK(11,4),WID(11,2)
      COMMON /MODEL7/ NPATCH,GINVEN,VENDV,DINVEN,JRATE,FTOTAL,FGRD,
     *FPAT,FWID,NGRD,GD,LN,GRA,XTOTAL,XGRD,XWID,CNT,CNTR,VCODE
      COMMON /MODEL8/ GRD,THK,LEN,TYP,DDATE,ORIQTY,SS1,SS2,CODE,QTY,
     *FRC,PRATE,PD,P,COMPL,NM,PROW,TROW,FILLED,GRADE,THICK,LENGTH,
     *WID
      COMMON /MODEL9/ TAIL,HEAD,LINKPR,MASTER,NAR,ROW,AHEAD,BEHIND,
     *IHEAD,ITAIL,ITEM,DLT,SIZE,LINKSR,TRNSFR,LSIZE
C
C   IF THE DESIRED GRADE WAS NOT FOUND IN INVENTORY, RETURN
C
      NWD=SS2
5     IF(ORIQTY.EQ.QTY)GO TO 15
C
C   IF QUANTITY WAS NOT COMPLETELY FILLED, CONTINUE
C
      IF(ORIQTY.GT.0.)GO TO 10
C
C  ORDER WAS COMPLETELY FILLED
C
C   COMPUTING PROCESSING TIME FOR THE JOINTER OPERATION
C
      PROCTM=NINT(QTY/PRATE(3))
C
C   COMPLETION TIME FOR THE JOINTER OPERATION
C
      DDATE(3)=DDATE(3)+PROCTM
C
C   CALLING SLAVE ROUTINE TO SCHEDULE TASK ON ONE OF THE MACHINES ON
C   THE JOINTER OPERATION
C
      CALL TRNSMT(1,3,PROCTM,SS1,THK,LEN,SS2,CODE,DDATE(3),
     *TROW,PRATE(3),NM,PROW)
      PROW(1)=TROW
      NM=1
C
C   COMPUTING PROCESSING TIME FOR THE PATCHER OPERATIN
C
      PROCTM=NINT(QTY/PRATE(2))
C
C   COMPLETION TIME FOR THE PATCHER OPERATION
```

```
C
      DDATE(2)=DDATE(3)+PROCTM
C
C     CALLING THE SLAVE ROUTINE FOR SCHEDULING THE TASK ON ONE OF THE
C     MACHINES OF THE PATCHER OPERATION
C
      CALL TRNSMT(1,2,PROCTM,SS1,THK,LEN,SS2,CODE,DDATE(2),
     *TROW,PRATE(2),NM,PROW)
      PD=1
      COMPL=DDATE(2)
      SS2=1
      CALL INVDEV
      FILLED=.TRUE.
      RETURN
C
C
C     ORDER WAS NOT COMPLETELY FILLED
C
10    QTY=QTY-ORIQTY
C
C     COMPUTING PROCESSING TIME FOR THE JOINTER OPERATION
C
      PROCTM=NINT(QTY/PRATE(3))
C
C     COMPLETION TIME FOR THE JOINTER OPERATION
C
      DDATE(3)=DDATE(3)+PROCTM
C
C     CALLING SLAVE ROUTINE FOR SCHEDULING A TASK ON ONE OF THE MACHINES
C     OF THE JOINTER OPERATION
C
      CALL TRNSMT(1,3,PROCTM,SS1,THK,LEN,SS2,CODE,DDATE(3),
     *TROW,PRATE(3),NM,PROW)
      PROW(1)=TROW
      NM=1
C
C     PROCESSING TIME FOR THE PATCHER OPERATION
C
      PROCTM=NINT(QTY/PRATE(2))
C
C     COMPLETION TIME FOR THE PATCHER OPERATION
C
      DDATE(2)=DDATE(3)+PROCTM
C
C     CALLING SLAVE ROUTINE FOR SCHEDULING A TASK ON ONE OF THE MACHINES
C     OF THE PATCHER OPERATION
C
      CALL TRNSMT(1,2,PROCTM,SS1,THK,LEN,SS2,CODE,DDATE(2),
     *TROW,PRATE(2),NM,PROW)
      PD=1
      COMPL=DDATE(2)
      SS2=1
      CALL INVDEV
      SS2=NWD
      QTY=ORIQTY
C
C     IF DESIRED INVENTORY HAS BEEN EXHAUSTED, RETURN
C
      IF(LSIZE(CODE).EQ.0)RETURN
      CALL NFIND(DDATE(3),CODE,ORIQTY)
```

```
      GO TO 5
15    CONTINUE
      RETURN
      END
```

```
C
C    FEB. 21.1984
C
C    THIS SUBROUTINE COMPUTES THE QUANTITIES FOR THE OTHER VENEER GRADES
C    THAT RESULT WHEN CREATING A TASK ON EITHER THE PATCHER OR DRYER
C    OPERATIONS.  IT ALSO USES SUB. FILE TO STORE THESE GRADES INTO
C    DRY INVENTORY.
C
C    AUTHOR:  ERIC SCHILD
C
      SUBROUTINE INVDEV
      INTEGER NPATCH(9.4.4,3),FTOTAL.FGRD(3).GINVEN(2.6.4,5.3).
     *FPAT(3).NGRD(3),GRA(3,2).FWID(3,4).XTOTAL,XGRD(3).XWID(3.4).
     *GD(10.2).DINVEN(11.4.5.3).COMPL.GRD.THK.LEN.TYP.DDATE(3).
     *SS1,SS2,CODE.PD.P.TAIL(660).HEAD(660).LINKPR(1500).NAR.ROW.
     *LINKSR(1500),MASTER(1500.3).AHEAD.BEHIND.IHEAD.ITAIL.ITEM.SIZE.
     *NROWS.DLT(75).LSIZE(660).TRNSFR(3).PROW(26),NM.TROW.LN.
     *LENGTH(11).VENDV(8.4.4).CNT.CNTR.VCODE
      REAL QTY.ORIQTY.FRC(11.2).PRATE(3).JRATE(4.4)
      LOGICAL FILLED
      BYTE GRADE(11.4).THICK(11.4).WID(11,2)
      COMMON /MODEL7/ NPATCH,GINVEN,VENDV.DINVEN.JRATE,FTOTAL.FGRD.
     *FPAT.FWID.NGRD.GD.LN.GRA.XTOTAL.XGRD.XWID.CNT.CNTR.VCODE
      COMMON /MODEL8/ GRD,THK,LEN,TYP,DDATE.ORIQTY.SS1.SS2.CODE,QTY.
     *FRC.PRATE.PD.P.COMPL.NM.PROW,TROW.FILLED.GRADE.THICK,LENGTH.
     *WID
      COMMON /MODEL9/ TAIL.HEAD,LINKPR,MASTER.NAR,ROW.AHEAD.BEHIND.
     *IHEAD.ITAIL,ITEM.DLT.SIZE.LINKSR.TRNSFR.LSIZE
C
C    TIME VENEER IS PLACED IN DRY INVENTORY
C
      TRNSFR(1)=COMPL
      TRNSFR(3)=TROW
      DO 10 I=1.8
C
C    IF GRADE IS EQUAL TO DESIRED GRADE OR GRADE FRACTION IS ZERO. SKIP
C
      DO 5 J=1.NGRD(CNT)
      IF(GD(I.PD).EQ.GRA(CNT,J))GO TO 10
5     CONTINUE
      IF(FRC(GD(I.PD),PD).EQ.0.) GO TO 10
C
C    DETERMINING DRY INVENTORY CODE TO BE USED IN STORING THE NEW
C    INVENTORY IN THE CORRECT LOCATION OF THE LINKED STORAGE ARRAY.
C
      INDEX=DINVEN(GD(I.PD).THK.LEN.SS2)
C
C    COMPUTING VENEER QUANTITY FOR EACH GRADE
C
      TRNSFR(2)=NINT(QTY*FRC(GD(I.PD).PD)*100.)
      CALL FILE(INDEX)
10    CONTINUE
      RETURN
      END
```

```
C
C    FEB. 21.1984
C
C    THIS SUBPROGRAM IS USED TO DEVELOP THE DESIRED VENEER GRADE
C    FROM GREEN INVENTORY BY CREATING A DRYER TASK.  IT ACCOMPLISHES
C    THIS BY PROMPTING THE USER FOR THE GREEN VENEER CODE NECESSARY
C    IN FILLING THE CURRENT ORDER.  IT ALSO ALLOWS THE USER TO OVER-
C    RIDE THE SYSTEM IN SPECIFYING THE TASK QUANTITY.
C
C    AUTHOR:  ERIC SCHILD
C
      SUBROUTINE DRYER
      INTEGER NPATCH(9.4.4,3).FTOTAL,FGRD(3),GINVEN(2,6,4.5.3),
     *FPAT(3),NGRD(3),GRA(3,2).FWID(3,4),XTOTAL,XGRD(3),XWID(3,4),
     *GD(10.2).DINVEN(11.4.5,3),COMPL.GRD.THK.LEN.TYP.DDATE(3).
     *SS1.SS2.CODE.PD.P.TAIL(660).HEAD(660),LINKPR(1500).NAR.ROW.
     *LINKSR(1500).MASTER(1500.3).AHEAD.BEHIND.IHEAD.ITAIL.ITEM.SIZE.
     *NROWS.DLT(75).LSIZE(660),TRNSFR(3),PROW(26).NM.TROW.LN,DCOD.
     *LENGTH(11).VENDV(8.4.4),CNT.CD.NR.NRR.NT.CNTR.VCODE,G.T.
     *L.W.GRNINV.N(2).PROCTM.NMBR.R(26)
      REAL QTY.SQTY.TQTY.TIMA,ORIQTY.FRC(11.2),PRATE(3).JRATE(4.4).
     *TIMC,AMT
      LOGICAL FILLED.ANS
      BYTE GRADE(11.4).THICK(11,4),WID(11,2),GR(4),TH(4),WI(2).
     *WIDTH(6.3)
      COMMON /MODEL7/ NPATCH.GINVEN,VENDV.DINVEN.JRATE.FTOTAL,FGRD.
     *FPAT.FWID.NGRD.GD.LN.GRA.XTOTAL,XGRD.XWID,CNT,CNTR.VCODE
      COMMON /MODEL8/ GRD.THK.LEN.TYP.DDATE.ORIQTY,SS1,SS2.CODE,QTY.
     *FRC.PRATE,PD,P.COMPL.NM.PROW.TROW,FILLED,GRADE,THICK,LENGTH,
     *WID
      COMMON /MODEL9/ TAIL.HEAD.LINKPR.MASTER.NAR.ROW.AHEAD.BEHIND.
     *IHEAD.ITAIL.ITEM.DLT.SIZE.LINKSR.TRNSFR.LSIZE
      DATA WIDTH/'F'.'A','C','E',' ',' '.'B','A','C','K',' ',' ',
     *'C','E','N','T','E','R'/
C
C   IF ORDER IS FILLED. RETURN
C
      NR=0
5     IF(ORIQTY.LE.0.)GO TO 90
      IERR=0
      AMT=ORIQTY
C
C    IF ORDER IS FULL WIDTH. CONTINUE
C
      IF(TYP.LT.4)GO TO 10
      CALL XBDTSK(GRD.THK.LEN,ORIQTY.NTP.G.T,L,W.AMT)
      GO TO 25
C
C    ORDER IS FULL WIDTH
C
C    PROMPTING USER FOR FULL WIDTH VENEER CODE.
C
10    WRITE(5.15) (WIDTH(M,TYP),M=1,6).(GRADE(GRD,M).M=1,4).
     *(THICK(THK,M).M=1,4).LENGTH(LEN),(WID(SS2,M),M=1,2),
     *ORIQTY
15    FORMAT('0'.9X.'ORDER TYPE: '.6A1/8X.'VENEER GRADE: '.4A1/
     *11X,'THICKNESS: '.4A1/14X.'LENGTH: '.I2/15X.'WIDTH: '.
     *2A1/6X.'ORDER QUANTITY: '.F6.2//4X.'FULL VENEER CODE: '.$)
      READ(5.20) NTP.G.T.L.W
20    FORMAT(5I1)
```

```
C
C    CHECKING TO SEE WHETHER CODE IS VIABLE.
C
35     CALL ERRCHK(NTP,G,T,L,W,IERR,DCOD)
C
C    IF IERR IS 1, CODE IS NOT VIABLE.
C
       IF(IERR.EQ.1)GO TO 5
       ORIQTY=AMT
C
C    READING FROM DRY VENEER DEVELOPMENT FILE.
C
       READ(3'DCOD) (FRC(GD(M,2),2),M=1,8),PRATE(1),(GR(M),M=1,4),
      *(TH(M),M=1,4),LENG,(WI(M),M=1,2)
C
C    IF DRY VENEER DEVELOPMENT FRACTION FOR DESIRED GRADE IS
C    ZERO, WRITE ERROR MESSAGE.
C
       IF(FRC(GRD,2).GT.0.)GO TO 35
       WRITE(5,30) NTP,G,T,L,W
30     FORMAT(T2,'THE CURRENT ORDER CANNOT BE FILLED USING THE
      *GREEN INVENTORY GRADE-',5I1)
       GO TO 5
C
C    CHECKING TO SEE IF USER WOULD LIKE TO SPECIFY HIS
C    OWN QUANTITY.
C
35     WRITE (5,40)
40     FORMAT(T2,'SPECIFY QUANTITY (Y:N)? ',$)
       READ (5,45) ANS
45     FORMAT(A1)
C
C    IF NO, COMPUTE REQUIRED QUANTITY
C
       IF(ANS.EQ.'Y')GO TO 50
       QTY=ORIQTY/FRC(GRD,2)
       ORIQTY=0.
       GO TO 60
C
C    PROMPTING USER FOR ORDER QUANTITY.
C
50     WRITE(5,55)
55     FORMAT(5X,'VENEER QUANTITY: ',$)
       ACCEPT*,QTY
       IF(QTY.EQ.0.)GO TO 5
60     IF(QTY.GE.3.)GO TO 65
       QTY=3.
C
C    IF TASK QUANTITY EXCEEDS 30 UNITS, BREAK TASK UP INTO SMALLER
C    SUBTASKS
C
65     TQTY=QTY
       TIMA=TQTY/30.
       TIMB=IFIX(TIMA)
       TIMC=TIMA-FLOAT(TIMB)
       NMBR=NINT(TIMA)
       DO 75 J=1,NMBR
       QTY=30.
       IF(J.LT.NMBR)GO TO 70
C
```

```
C   NO TASK CAN BE LESS THAN 3 UNITS
C
      QTY=AMAX1(TIMC*30.,3.)
C
C   COMPUTING PROCESSING TIME ON THE DRYER OPERATION
C
70    PROCTM=NINT(QTY/PRATE(1))
C
      LEN=L
      PROW(1)=-1
      NM=1
C
C   CALLING SLAVE ROUTINE FOR SCHEDULING A TASK ON ONE OF THE MACHINES
C   OF THE DRYER OPERATION
C
      CALL TRNSMT(1,4,PROCTM,GRD,THK,LEN,W,DCOD,DDATE(1),
     *TROW,PRATE(1),NM,PROW)
      NR=NR+1
      R(NR)=TROW
      PD=2
      COMPL=DDATE(1)
      SS2=W
      CNT=1
      NGRD(1)=1
      GRA(1,1)=GRD
      CALL INVDEV
75    CONTINUE
      DO 80 J=1,NR
      PROW(J)=R(J)
80    CONTINUE
C
C   RECOMPUTING ORDER QUANTITY
C
      QTY=TQTY*FRC(GRD,2)
      NT=1
      NRR=0
      CODE=DINVEN(GRD,THK,LEN,SS2)
      CD=THK*100+LEN*10+SS2
      WRITE(7,85) DDATE(1),QTY,CODE,CD,NT,NRR
85    FORMAT(I2,I6,F6.2,2I4,2I2)
      ORIQTY=ORIQTY-QTY
      GO TO 5
90    CONTINUE
      NM=NR
      RETURN
      END
```

```
C
C     APRIL 7, 1984
C
C     THIS SUBPROGRAM IS CALLED BY SUBROUTINE DRYER TO PROMPT THE USER
C     FOR THE GREEN VENEER CODE TO BE USED IN FILLING NARROW WITH VENEER
C     ORDERS.
C
C     AUTHOR:  ERIC R. SCHILD
C
      SUBROUTINE XBDTSK(G,T,L,Q,A,B,C,D,H,E)
      INTEGER NPATCH(9,4,4,3),FTOTAL,FGRD(3),GINVEN(2,6,4,5,3),
     *FPAT(3),NGRD(3),GRA(3,2),FWID(3,4),XTOTAL,XGRD(3),XWID(3,4),
     *GD(10,2),DINVEN(11,4,5,3),COMPL,GRD,THK,LEN,TYP,DDATE(3),
     *SS1,SS2,CODE,PD,P,TAIL(660),HEAD(660),LINKPR(1500),NAR,ROW,
     *LINKSR(1500),MASTER(1500,3),AHEAD,BEHIND,IHEAD,ITAIL,ITEM,SIZE,
     *NROWS,DLT(75),LSIZE(660),TRNSFR(3),PROW(26),NM,TROW,LN,
     *LENGTH(11),VENDV(8,4,4),CNT,G,T,L,H,A,B,C,D,N(5),CNTR,VCODE
      REAL QTY,E,Q,V(3),F(4,3),ORIQTY,FRC(11,2),PRATE(3),JRATE(4,4)
      LOGICAL FILLED
      BYTE GRADE(11,4),WIDTH(4),THICK(11,4),WID(11,2)
      COMMON /MODEL7/ NPATCH,GINVEN,VENDV,DINVEN,JRATE,FTOTAL,FGRD,
     *FPAT,FWID,NGRD,GD,LN,GRA,XTOTAL,XGRD,XWID,CNT,CNTR,VCODE
      COMMON /MODEL8/ GRD,THK,LEN,TYP,DDATE,ORIQTY,SS1,SS2,CODE,QTY,
     *FRC,PRATE,PD,P,COMPL,NM,PROW,TROW,FILLED,GRADE,THICK,LENGTH,
     *WID
      COMMON /MODEL9/ TAIL,HEAD,LINKPR,MASTER,NAR,ROW,AHEAD,BEHIND,
     *IHEAD,ITAIL,ITEM,DLT,SIZE,LINKSR,TRNSFR,LSIZE
C
      DATA WIDTH/'X','B','N','D'/
      DATA F/2.,2.25,2.5,3.,1.,1.125,1.25,1.5,.67,.75,.83,1./
      DATA N/2,2,2,3,1/
C
C     PROMPTING THE USER FOR THE GREEN VENEER CODE.
C
      WRITE(5,5) (WIDTH(M),M=1,4),(GRADE(GRD,M),M=1,4),
     *(THICK(THK,M),M=1,4),LENGTH(L),(WID(3,M),M=1,2),
     *Q
5     FORMAT('0',9X,'ORDER TYPE: ',4A1/8X,'VENEER GRADE: ',4A1/
     *11X,'THICKNESS: ',4A1/14X,'LENGTH: ',I2/15X,'WIDTH: ',
     *2A1/6X,'ORDER QUANTITY: ',F6.2)
C
C     COMPUTING EQUIVALENT VENEER QUANTITIES IF THE USER DECIDES
C     TO USE 4', 8', OR 12' VENEER.
C
      DO 10 I=1,3
      V(I)=Q*F(L,I)
10    CONTINUE
C
C     LISTING ALTERNATIVE VENEER SOURCES AND REQUIRED QUANTITIES
C
      WRITE(5,15) (V(M),M=1,3)
15    FORMAT(' ',10X,'ACTUAL VENEER REQTS'/14X,'USING 4:',F6.2/20X,'8:',
     *F6.2/19X,'12:',F6.2//4X,'XBND VENEER CODE: ',$)
      READ(5,20) A,B,C,D,H
20    FORMAT(5I1)
      E=V(N(D))
      RETURN
      END
```

```
C
C     APRIL 7, 1984
C
C     THIS SUBPROGRAM IS USED TO VALIDATE THE USER SPECIFIED GREEN
C     INVENTORY CODE.  IF THE CODE ISN'T VIABLE, AN ERROR MESSAGE IS
C     WRITTEN AND THE USER MUST ENTER ANOTHER GREEN INVENTORY CODE.
C
C     AUTHOR:  ERIC R. SCHILD
C
      SUBROUTINE ERRCHK(A,B,C,D,H,ERR,DCOD)
      INTEGER NPATCH(9,4,4,3),FTOTAL,FGRD(3),GINVEN(2,6,4,5,3),
     *FPAT(3),NGRD(3),GRA(3,2),FWID(3,4),XTOTAL,XGRD(3),XWID(3,4),
     *GD(10,2),DINVEN(11,4,5,3),COMPL,GRD,THK,LEN,TYP,DDATE(3),
     *SS1,SS2,CODE,PD,P,TAIL(660),HEAD(660),LINKPR(1500),NAR,ROW,
     *LINKSR(1500),MASTER(1500,3),AHEAD,BEHIND,IHEAD,ITAIL,ITEM,SIZE,
     *NROWS,DLT(75),LSIZE(660),TRNSFR(3),PROW(26),NM,TROW,LN,
     *LENGTH(11),VENDV(8,4,4),CNT,ERR,DCOD,A,B,C,D,H,CNTR,VCODE
      REAL QTY,ORIQTY,FRC(11,2),PRATE(3),JRATE(4,4)
      LOGICAL FILLED
      BYTE GRADE(11,4),THICK(11,4),WID(11,2)
      COMMON /MODEL7/ NPATCH,GINVEN,VENDV,DINVEN,JRATE,FTOTAL,FGRD,
     *FPAT,FWID,NGRD,GD,LN,GRA,XTOTAL,XGRD,XWID,CNT,CNTR,VCODE
      COMMON /MODEL8/ GRD,THK,LEN,TYP,DDATE,ORIQTY,SS1,SS2,CODE,QTY,
     *FRC,PRATE,PD,P,COMPL,NM,PROW,TROW,FILLED,GRADE,THICK,LENGTH,
     *WID
      COMMON /MODEL9/ TAIL,HEAD,LINKPR,MASTER,NAR,ROW,AHEAD,BEHIND,
     *IHEAD,ITAIL,ITEM,DLT,SIZE,LINKSR,TRNSFR,LSIZE
C
C     SPLIT PEEL OR REGULAR
C
      IF((A.GE.1).AND.(A.LE.2))GO TO 4
      WRITE(5,3) A
3     FORMAT(T2,'THE TYPE CODE IS OUT OF RANGE-',I2)
C
C     CHECKING THE GRADE CODE.
C
4     IF((B.GE.1).AND.(B.LE.6))GO TO 10
      WRITE(5,5) B
5     FORMAT(T2,'THE GRADE CODE IS OUT OF RANGE-',I2)
      ERR=1
C
C     CHECKING THE THICKNESS CODE.
C
10    IF(C.EQ.THK)GO TO 17
      WRITE(5,15) C
15    FORMAT(T2,'THE THICKNESS CODE IS OUT OF RANGE-',I2)
      ERR=1
C
C     CHECKING THE LENGTH CODE.
C
17    IF(TYP.LT.4)GO TO 20
      IF((D.GE.1).AND.(D.LE.5))GO TO 30
      WRITE(5,18) D
18    FORMAT(T2,'THE LENGTH CODE IS OUT OF RANGE-',I2)
      ERR=1
      GO TO 30
20    IF(D.EQ.LEN)GO TO 30
      WRITE(5,25) D
25    FORMAT(T2,'THE LENGTH CODE IS OUT OF RANGE-',I2)
      ERR=1
```

```
C
C   CHECKING THE WIDTH CODE.
C
30     IF((H.GE.1).AND.(H.LE.3))GO TO 40
       WRITE(5,35) H
35     FORMAT(I2,'THE WIDTH CODE IS OUT OF RANGE-',I2)
       ERR=1
40     DCOD=GINVEN(A,B,C,D,H)
C
C   IF INDEX IS ZERO, WRITE ERROR MESSAGE.
C
       IF(DCOD.GT.0)RETURN
       WRITE(5,45) A,B,C,D,H
45     FORMAT(I2,'THE CURRENT CODE-',5I1,' DOES NOT EXIST IN
     *DRYDEV.DAT')
       ERR=1
       RETURN
       END
```

A.2.2 Variable Definitions and Listing of Scheduling Program

AHEAD - Dummy variable used to retain identity of next row of linked starage array.

BEHIND - Dummy varialbe used to retain identity of previous row of linked storage array.

COD - Veneer inventory code.

CPTY(i) - Machine capacity of ith operation. i=1,4.

DUE - Due date of veneer task.

G - Veneer grade code.

HEAD(i) or HED(j) - Head pointer. i=1,72, j=1,1500.

IDLE - Amount of idle time.

IHEAD - Dummy variable used to retain identity of HEAD (i).

INDX(i,j) - Index code for jth machine on ith operation; used to sschedule task in the appropriate position of the linked storage array. i=1,4, j=1,18.

ITAIL - Dummy variable used to retain identity of TAIL(i).

L - Veneer length code.

LINKPR(i) and LNKPR(j) - Predecessor pointers for the two respective linked lists. i=1,1500, j=1,2000.

LINKSR(i) and LNKSR(j) - Successor pointer for the two respective linked lists. i=1,1500, j=1,2000.

LSIZE(i) and LSIZ - Number of records in list. i=1,72.

MACH - Machine number.
MASTER(i,j) - Primary linked storage array used to retain the task schedules for each machine of each operation. It stores the start and completion times of the task in addition to the processing rate and identification code. i=1,1500, j=1,4.

MSTR(i) - Secondary linked storage array used to identify the processing path of each task in the primary linked storage array. i=1,2000. NAR(i) - Next available row in the ith linked storage array. i=1,2.

OPER - Operation task is to be processed.

OVRLAP - Amount of time one task (being scheduled) overlaps another.

POLICY(i,j,k,l) - Machine restriction array for machine i in operation j. i=1,18, j=1,4, k=1,7, l=1,4.

PRATE - Production rate for veneer task on OPER.

PROC - Total processing time required for veneer task.

PROW(i) - List of development codes indicating the tasks used to create dry veneer.

PRTY(i) - Task allocation array used to assign tasks to the machine with the smallest total processing time. i=1,18.

REQTIM - The required time task is needed on a succeeding operation.

ROW, R, RW, and RW1 - Row number of either primary or secondary linked list.

T - Veneer thickness code.

TAIL(i) or TAL(j) - Tail pointer. i=1,72, j=1,1500.

TIME(i,j) - Total amount of processing assigned to ith machine of jth operation. i=1,18, j=1,4.

TRNSFR(i) - Transfer array used to store the completion and start times, and the processing rate and task identification code in the linked storage array. i=1,5.

TROW - Development code indicating the task used to create dry veneer.

W - Veneer width code.

```
C
C   APRIL 30, 1984
C
C   THIS PROGRAM ACTS AS THE MAIN DRIVER FOR SCHEDULING TASKS
C   CREATED BY THE MASTER ROUTINE.  IT SPECIFICALLY INTERCEPTS DATA
C   FROM AND SENDS DATA TO THE MASTER ROUTINE.
C
C   AUTHOR:  ERIC SCHILD
C
      PROGRAM RECV
      INTEGER TIME(18,4),POLICY(18,4,7,4),PRTY(18,4),CPTY(4),OPER,
     *FLAG,G,T,L,W,COD,DUE,PROC,MACH,PRATE,J(15),K(13),PROW(26),
     *SIZE,NM,N,TROW,INDX(4,18),MASTER(1500,4),MSTR(2000),
     *LINKPR(1500),LINKSR(1500),LNKPR(2000),LNKSR(2000),ROW,TRNSFR(5),
     *NAR(2),TAIL(72),HEAD(72),LSIZE(72),TAL(1500),HED(1500),
     *LSIZ(1500),BEHIND,AHEAD,ITAIL,POLI,NUM,LIST,INDEX,R,
     *OVRLAP,IDLE,REQTIM,RW,RW1
      COMMON /MODL10/ TIME,POLICY,PRTY,CPTY,OPER,FLAG,G,T,L,W,COD,
     *DUE,PROC,INDX,PRATE,MACH,PROW,TROW,NM
      COMMON /MODL11/ MASTER,MSTR,LINKPR,LINKSR,LNKPR,LNKSR,HEAD,
     *TAIL,HED,TAL,LSIZE,LSIZ,TRNSFR,NAR
      DATA FILE/5RPHAS2/
C
      CALL INITK2
C
C    WAITING FOR THE NEXT TASK TO BE SENT FROM THE MASTER ROUTINE
C
5     CALL WAITFR(33)
      CALL RECEIV(FILE,J)
      CALL CLREF(33)
      FLAG=J(3)
      OPER=J(4)
      PROC=J(5)
      G=J(6)
      T=J(7)
      L=J(8)
      W=J(9)
      COD=J(10)
      DUE=J(11)
      PRATE=J(12)
      NM=J(13)
      SIZE=NINT(FLOAT(NM)/13.)
      N=1
      DO 20 I=1,SIZE
      CALL SETEF(37)
      CALL WAITFR(33)
      CALL RECEIV(FILE,J)
      CALL CLREF(33)
      DO 10 M=3,15
      PROW(N)=J(M)
      N=N+1
10    CONTINUE
20    CONTINUE
      CALL SETEF(37)
C
C    IF FLAG IS LESS THAN 0, CALL SUBROUTINE TALLY AND TERMINATE PROGRAM
C
      IF(FLAG.LT.0)GO TO 30
      CALL MASGN
C
```

```
      K(1)=TROW
      CALL CLREF(35)
      CALL SEND(FILE,K,35)
C
      GO TO 5
30    CALL TALLY
      STOP
      END
```

```
C
C     JUNE 4, 1984
C
C     THIS SUBPROGRAM IS USED TO INITIALIZE THE LINKED STORAGE ARRAYS
C     WHICH WILL BE USED TO STORE THE TASK SCHEDULES FOR EACH MACHINE IN
C     EACH OPERATION.
C
C     AUTHOR:  ERIC SCHILD
C
      SUBROUTINE INITK2
      INTEGER TIME(18,4),POLICY(18,4,7,4),PRTY(18,4),CPTY(4),OPER,
     *FLAG,G,T,L,W,COD,DUE,PROC,MACH,PRATE,PROW(26),
     *SIZE,NM,N,TROW,INDX(4,18),MASTER(1500,4),MSTR(2000),
     *LINKPR(1500),LINKSR(1500),LNKPR(2000),LNKSR(2000),ROW,TRNSFR(5),
     *NAR(2),TAIL(72),HEAD(72),LSIZE(72),TAL(1500),HED(1500),
     *LSIZ(1500),BEHIND,AHEAD,ITAIL,POLI,NUM,LIST,INDEX,R,
     *OVRLAP,IDLE,REQTIM,RW,RW1
      COMMON /MODL10/ TIME,POLICY,PRTY,CPTY,OPER,FLAG,G,T,L,W,COD,
     *DUE,PROC,INDX,PRATE,MACH,PROW,TROW,NM
      COMMON /MODL11/ MASTER,MSTR,LINKPR,LINKSR,LNKPR,LNKSR,HEAD,
     *TAIL,HED,TAL,LSIZE,LSIZ,TRNSFR,NAR
C
      DATA CPTY,NAR/6,18,5,7,1,1/
C
C     INITIALIZING LINKS
C
      DO 10 ROW=1,1500
      LINKPR(ROW)=0
      LINKSR(ROW)=ROW+1
10    CONTINUE
      LINKSR(1500)=0
C
C     INITIALIZING ATTRIBUTES OF LIST
C
      DO 20 LIST=1,72
      HEAD(LIST)=0
      TAIL(LIST)=0
      LSIZE(LIST)=0
20    CONTINUE
C
C     INITIALIZING LINKS
C
      DO 25 ROW=1,2000
      LNKPR(ROW)=0
      LNKSR(ROW)=ROW+1
25    CONTINUE
      LNKSR(2000)=0
C
C     INITIALIZING ATTRIBUTES OF LIST
C
      DO 30 LIST=1,1500
      HED(LIST)=0
      TAL(LIST)=0
      LSIZ(LIST)=0
30    CONTINUE
C
C     SETTING UP LIST INDEX ARRAY, MACHINE PRIORITY ARRAY, AND
C     PROCESSING TIME ARRAY
C
      LIST=1
```

```
      DO 40 J=1,4
      DO 35 M=1,18
      INDX(J,M)=LIST
      PRTY(M,J)=M
      TIME(M,J)=0
      LIST=LIST+1
35    CONTINUE
40    CONTINUE
C
C   SETTING UP THE MACHINE RESTRICTIONS FOR THE DRYER AND JOINTER
C   OPERATIONS
C
      CALL ASSIGN(3,'SY:JPOLI.DAT')
      CALL ASSIGN(4,'SY:DPOLI.DAT')
      DO 60 J=3,4
      L=J
      INDEX=1
50    READ(L,55,END=60) ((POLICY(INDEX,L,M,K),M=1,7),K=1,4)
55    FORMAT(T2,28I1)
      INDEX=INDEX+1
      GO TO 50
60    CONTINUE
      CALL CLOSE(3)
      CALL CLOSE(4)
      RETURN
      END
```

```
C
C   JUNE 5, 1984
C
C   THIS SUBPROGRAM IS THE LAST ROUTINE CALLED FROM PROGRAM RECV
C   BEFORE THE PROGRAM IS TERMINATED.  THIS SUBROUTINE WRITES THE
C   CONTENTS OF THE THREE SCHEDULED LISTS FROM THE MASTER ARRAY ON
C   TO PERMANENT FILES.
C
C   AUTHOR:  ERIC SCHILD
C
      SUBROUTINE TALLY
      INTEGER TIME(18,4),POLICY(18,4,7,4),PRTY(18,4),CPTY(4),OPER,
     *FLAG,G,T,L,W,COD,DUE,PROC,MACH,PRATE,PROW(26),
     *SIZE,NM,N,TROW,INDX(4,18),MASTER(1500,4),MSTR(2000),
     *LINKPR(1500),LINKSR(1500),LNKPR(2000),LNKSR(2000),ROW,TRNSFR(5),
     *NAR(2),TAIL(72),HEAD(72),LSIZE(72),TAL(1500),HED(1500),
     *LSIZ(1500),BEHIND,AHEAD,ITAIL,POLI,NUM,LIST,INDEX,R,
     *OVRLAP,IDLE,REQTIM,RW,RW1
      COMMON /MODL10/ TIME,POLICY,PRTY,CPTY,OPER,FLAG,G,T,L,W,COD,
     *DUE,PROC,INDX,PRATE,MACH,PROW,TROW,NM
      COMMON /MODL11/ MASTER,MSTR,LINKPR,LINKSR,LNKPR,LNKSR,HEAD,
     *TAIL,HED,TAL,LSIZE,LSIZ,TRNSFR,NAR
C
      DO 70 OPER=2,4
      GO TO (10,20,30) OPER-1
10    CALL ASSIGN(1,'SY:PATTSK.DAT')
      GO TO 40
20    CALL ASSIGN(1,'SY:JOITSK.DAT')
      GO TO 40
30    CALL ASSIGN(1,'SY:DRYTSK.DAT')
40    DO 60 MACH=1,CPTY(OPER)
      LIST=INDX(OPER,MACH)
      CNT=1
      ROW=HEAD(LIST)
C
C   IF COUNTER EXCEEDS MAXIMUM SIZE OF LIST, RETURN
C
45    IF(CNT.GT.LSIZE(LIST))GO TO 60
      WRITE(1,50) (MASTER(ROW,I),I=1,4),MACH
50    FORMAT(T2,2I6,I4,I6,I3)
      CNT=CNT+1
      ROW=LINKSR(ROW)
      GO TO 45
60    CONTINUE
      CALL CLOSE(1)
70    CONTINUE
      RETURN
      END
```

```
C
C   FEB. 22, 1984
C
C   THIS SUBPROGRAM USES THE MACHINE RESTRICTION FILES TO DETERMINE
C   WHETHER A TASK CAN BE ASSIGNED TO A MACHINE OF A GIVEN OPERATION.
C   IF NO RESTRICTIONS EXIST, THEN THE TASK IS ASSIGNED. OTHERWISE,
C   ANOTHER MACHINE IS CHECKED.
C
C   AUTHOR:  ERIC SCHILD
C
      SUBROUTINE MASGN
      INTEGER TIME(18,4),POLICY(18,4,7,4),PRTY(18,4),CPTY(4),OPER,
     *FLAG,G,T,L,W,COD,DUE,PROC,MACH,PRATE,PROW(26),
     *SIZE,NM,N,TROW,INDX(4,18),MASTER(1500,4),MSTR(2000),
     *LINKPR(1500),LINKSR(1500),LNKPR(2000),LNKSR(2000),ROW,TRNSFR(5),
     *NAR(2),TAIL(72),HEAD(72),LSIZE(72),TAL(1500),HED(1500),
     *LSIZ(1500),BEHIND,AHEAD,ITAIL,POLI,NUM,LIST,INDEX,R,
     *OVRLAP,IDLE,REQTIM,RW,RW1
      COMMON /MODL10/ TIME,POLICY,PRTY,CPTY,OPER,FLAG,G,T,L,W,COD,
     *DUE,PROC,INDX,PRATE,MACH,PROW,TROW,NM
      COMMON /MODL11/ MASTER,MSTR,LINKPR,LINKSR,LNKPR,LNKSR,HEAD,
     *TAIL,HED,TAL,LSIZE,LSIZ,TRNSFR,NAR
C
C   IF TASK CANNOT BE ASSIGNED TO THE CURRENT MACHINE, INCREMENT MACHINE
C   LOOP COUNTER AND CHECK AGAIN.
C
      MACH=1
15    IF(MACH.LE.CPTY(OPER)) GO TO 18
C
      WRITE(5,16)
16    FORMAT(T2,'TASK CANNOT BE ASSIGNED UNDER THE CURRENT'/
     *T2,'POLICY RESTRICTIONS')
      STOP
C
C   CHECKING TO SEE IF CURRENT TASK CAN BE ASSIGNED TO CURRENT MACHINE
C   USING THE MACHINE RESTRICTION VARIABLES
C
18    DO 45 POLI=1,4
      DO 40 NUM=1,POLICY(PRTY(MACH,OPER),OPER,1,POLI)
      GO TO (20,25,30,35) POLI
C
C   CHECKING GRADE POLICY
C
20    IF(G.NE.POLICY(PRTY(MACH,OPER),OPER,NUM+1,POLI)) GO TO 40
      MACH=MACH+1
      GO TO 15
C
C   CHECKING THICKNESS POLICY
C
25    IF(T.NE.POLICY(PRTY(MACH,OPER),OPER,NUM+1,POLI)) GO TO 40
      MACH=MACH+1
      GO TO 15
C
C   CHECKING LENGTH POLICY
C
30    IF(L.NE.POLICY(PRTY(MACH,OPER),OPER,NUM+1,POLI)) GO TO 40
      MACH=MACH+1
      GO TO 15
C
C   CHECKING WIDTH POLICY
```

```
C
35    IF(W.NE.POLICY(PRTY(MACH,OPER),OPER,NUM+1,POLI)) GO TO 40
      MACH=MACH+1
      GO TO 15
40    CONTINUE
45    CONTINUE
C
C   ASSIGNING CURRENT TASK TO CURRENT MACHINE
C
C   UPDATING TOTAL PROCESSING TIME COUNTER
C
      TIME(PRTY(MACH,OPER),OPER)=TIME(PRTY(MACH,OPER),OPER)+PROC
      LIST=INDX(OPER,PRTY(MACH,OPER))
      TRNSFR(1)=DUE
      TRNSFR(2)=DUE-PROC
      TRNSFR(3)=COD
      TRNSFR(4)=PRATE
      CALL FILEM1(LIST)
      CALL PRTIZE
      RETURN
      END
```

```
C
C   FEB. 22. 1984
C
C   THIS SUBPROGRAM IS USED TO ENSURE THAT SUB MASGN WILL ALWAYS
C   ASSIGN THE TASK BY CONSIDERING THE MACHINE WITH THE LOWEST
C   TOTAL PROCESSING TIME FIRST FOLLOWED BY THE MACHINE WITH THE
C   NEXT TO THE LOWEST TOTAL AND SO FORTH.   IT ACCOMPLISHES THIS
C   USING AN ARRAY NAMED 'PRTY' WHICH REORGANIZES THE
C   LIST OF MACHINES SO THAT THE MACHINE WITH THE SMALLEST
C   AMOUNT OF PROCESSING TIME ASSIGNED TO IT WILL BE CONSIDERED
C   FIRST WHEN ASSIGNING THE NEXT JOB IN THE TASK FILE.
C
C   AUTHOR:  ERIC SCHILD
C
      SUBROUTINE PRTIZE
      INTEGER TIME(18.4),POLICY(18.4.7.4),PRTY(18.4),CPTY(4).OPER.
     *FLAG,G,T,L.W.COD.DUE.PROC,MACH,PRATE,PROW(26).
     *SIZE.NM.N.TROW.INDX(4.18).MASTER(1500.4).MSTR(2000).
     *LINKPR(1500),LINKSR(1500),LNKPR(2000),LNKSR(2000),ROW.TRNSFR(5).
     *NAR(2).TAIL(72).HEAD(72),LSIZE(72),TAL(1500).HED(1500).
     *LSIZ(1500).BEHIND,AHEAD,ITAIL,POLI,NUM,LIST,INDEX,R.
     *OVRLAP.IDLE.REQTIM.RW.RW1
      LOGICAL SORTED
      COMMON /MODL10/ TIME,POLICY.PRTY.CPTY,OPER,FLAG.G.T,L,W.COD.
     *DUE.PROC,INDX,PRATE.MACH,PROW,TROW,NM
      COMMON /MODL11/ MASTER.MSTR.LINKPR,LINKSR.LNKPR.LNKSR.HEAD.
     *TAIL,HED,TAL.LSIZE,LSIZ,TRNSFR,NAR
C
C   BUBBLE SORT
C
      SORTED=.FALSE.
5     IF(SORTED)GO TO 15
      SORTED=.TRUE.
      DO 10 I=1,CPTY(OPER)-1
      IF(TIME(PRTY(I.OPER),OPER).LE.TIME(PRTY(I+1,OPER).OPER))
     *GO TO 10
      M=PRTY(I,OPER)
      PRTY(I.OPER)=PRTY(I+1,OPER)
      PRTY(I+1,OPER)=M
      SORTED=.FALSE.
10    CONTINUE
      GO TO 5
15    CONTINUE
      RETURN
      END
```

```
C
C    JUNE 5, 1984
C
C    THIS SUBPROGRAM IS USED TO STORE THE NEWLY SCHEDULED TASK IN THE
C    APPROPRIATE LIST OF THE MASTER STORAGE ARRAY.  IT IS NEARLY
C    IDENTICAL TO THE FILING SUBPROGRAM USED IN SIMLIB.
C
C    AUTHOR:  ERIC SCHILD
C
      SUBROUTINE FILEM1(LIST)
      INTEGER TIME(18,4),POLICY(18,4,7,4),PRTY(18,4),CPTY(4),OPER,
     *FLAG,G,T,L,W,COD,DUE,PROC,MACH,PRATE,PROW(26),
     *SIZE,NM,N,TROW,INDX(4,18),MASTER(1500,4),MSTR(2000),
     *LINKPR(1500),LINKSR(1500),LNKPR(2000),LNKSR(2000),ROW,TRNSFR(5),
     *NAR(2),TAIL(72),HEAD(72),LSIZE(72),TAL(1500),HED(1500),
     *LSIZ(1500),BEHIND,AHEAD,ITAIL,POLI,NUM,LIST,INDEX,R,
     *OVRLAP,IDLE,REQTIM,RW,RW1
      COMMON /MODL10/ TIME,POLICY,PRTY,CPTY,OPER,FLAG,G,T,L,W,COD,
     *DUE,PROC,INDX,PRATE,MACH,PROW,TROW,NM
      COMMON /MODL11/ MASTER,MSTR,LINKPR,LINKSR,LNKPR,LNKSR,HEAD,
     *TAIL,HED,TAL,LSIZE,LSIZ,TRNSFR,NAR
C
C    IF THE MASTER STORAGE ARRAY IS FULL, STOP THE PROGRAM
C
      IF(NAR(1).GT.0) GO TO 15
      WRITE(5,10)
10    FORMAT(5X,'MASTER STORAGE ARRAY OVERFLOW')
      STOP
C
C    IF THE LIST VALUE IS IMPROPER, STOP THE PROGRAM
C
15    IF((LIST.GE.1).AND.(LIST.LE.72))GO TO 20
      WRITE(5,18)
18    FORMAT(5X,I4,' IS AN IMPROPER VALUE FOR LIST')
      STOP
C
C    IF THIS IS NOT THE FIRST RECORD IN THIS LIST, CONTINUE
C
20    IF(LSIZE(LIST).EQ.0)GO TO 85
C
      ROW=HEAD(LIST)
      AHEAD=HEAD(LIST)
C
C    SEARCHING THE LIST FOR THE PROPER LOCATION TO INSERT THE TASK
C
C    IF THE DUE DATE OF CURRENT TASK IS GREATER THAN THE DUE DATE OF
C    THE CURRENT RECORD, ASSIGN TASK AHEAD OF CURRENT RECORD
C
30    IF(TRNSFR(1).LT.MASTER(ROW,1))GO TO 60
C
C    COMPUTING AVAILABLE TIME
C
      AVTIM=TRNSFR(1)-MASTER(ROW,1)
C
C    IF AVAILABLE TIME IS GREATER THAN OR EQUAL TO TOTAL PROCESSING FOR
C    CURRENT TASK, NO CONFLICTS EXIST
C
      IF(AVTIM.GE.PROC) GO TO 50
C
C    CURRENT TASK CONFLICTS WITH CURRENT RECORD
```

```
C
      OVRLAP=MASTER(ROW,1)-TRNSFR(2)
      CALL PSHBK1(ROW,OVRLAP)
C
50    ROW=NAR(1)
      NAR(1)=LINKSR(ROW)
      IF(NAR(1).GT.0)LINKPR(NAR(1))=0
      IF(AHEAD.NE.HEAD(LIST)) GO TO 55
C
C   INSERTING CURRENT TASK BEFORE FIRST RECORD IN THE LIST
C
      LINKPR(AHEAD)=ROW
      LINKSR(ROW)=AHEAD
      LINKPR(ROW)=0
      HEAD(LIST)=ROW
      GO TO 90
C
C   INSERTING CURRENT TASK BETWEEN THE PRECEEDING AND SUCCEEDING
C   RECORDS
C
55    LINKPR(ROW)=BEHIND
      LINKSR(BEHIND)=ROW
      LINKPR(AHEAD)=ROW
      LINKSR(ROW)=AHEAD
      GO TO 90
C
C   IF THE DUE DATE OF CURRENT TASK CONFLICTS WITH START TIME OF
C   CURRENT RECORD, PUSH BACK CURRENT RECORD AND PLACE CURRENT TASK
C   AHEAD OF CURRENT RECORD
C
60    IF(TRNSFR(1).LE.MASTER(ROW,2)) GO TO 80
C
C   COMPUTING OVER LAP
C
      OVRLAP=MASTER(ROW,1)-TRNSFR(2)
      CALL PSHBK1(ROW,OVRLAP)
C
      ROW=NAR(1)
      NAR(1)=LINKSR(ROW)
      IF(NAR(1).GT.0) LINKPR(NAR(1))=0
      IF(AHEAD.NE.HEAD(LIST)) GO TO 75
C
C   INSERTING CURRENT TASK BEFORE FIRST RECORD
C
      LINKPR(AHEAD)=ROW
      LINKSR(ROW)=AHEAD
      LINKPR(ROW)=0
      HEAD(LIST)=ROW
      GO TO 90
C
C   INSERTING CURRENT TASK BETWEEN THE PRECEEDING AND SUCCEEDING
C   RECORD
C
75    LINKPR(ROW)=BEHIND
      LINKSR(BEHIND)=ROW
      LINKPR(AHEAD)=ROW
      LINKSR(ROW)=AHEAD
      GO TO 90
C
C   CONTINUE SEARCHING, CONSIDER THE NEXT ROW
```

```
C
30      BEHIND=AHEAD
        AHEAD=LINKSR(AHEAD)
        ROW=AHEAD
C
C    IF THE LAST ROW CONSIDERED WAS NOT THE TAIL OF THE LIST, CONTINUE
C
        IF(AHEAD.GT.0)GO TO 30
        ROW=NAR(1)
        NAR(1)=LINKSR(ROW)
        IF(NAR(1).GT.0) LINKPR(NAR(1))=0
C
C    INSERTING AFTER LAST RECORD IN THE LIST
C
        ITAIL=TAIL(LIST)
        LINKPR(ROW)=ITAIL
        LINKSR(ITAIL)=ROW
        LINKSR(ROW)=0
        TAIL(LIST)=ROW
        GO TO 90
C
85      ROW=NAR(1)
        NAR(1)=LINKSR(ROW)
        IF(NAR(1).GT.0) LINKPR(NAR(1))=0
C
C    INSERTING THE FIRST RECORD IN THE LIST
C
        LINKSR(ROW)=0
        HEAD(LIST)=ROW
        TAIL(LIST)=ROW
C
C    TRANSFERING DATA TO MASTER ARRAY
C
90      DO 100 ITEM=1,4
        MASTER(ROW,ITEM)=TRNSFR(ITEM)
100     CONTINUE
        TROW=ROW
C
C    INCREMENTING LIST SIZE COUNTER
C
        LSIZE(LIST)=LSIZE(LIST)+1
C
C    CALLING FILEM2 TO SET UP THE SECONDARY LINKS
C
C    THIS WILL LINK THE CURRENT TASK TO ITS RAW MATERIAL SOURCES ON
C    PRECEEDING OPERATIONS
C
        DO 110 ITEM=1,NM
        TRNSFR(1)=PROW(ITEM)
        CALL FILEM2(ROW)
110     CONTINUE
        RETURN
        END
```

```
C
C   JUNE 5, 1984
C
C   THIS SUBPROGRAM PERFORMS THE SECONDARY LINKING FUNCTION SO THAT THE
C   MATERIAL FLOW CAN BE TRACED THROUGH EACH OPERATION.  LIKE SUBROUTINE
C   FILEM1, THIS SUBPROGRAM IS NEARLY IDENTICAL TO THE FILING SUBROUTINE
C   USED IN SIMLIB.
C
C   AUTHOR:  ERIC SCHILD
C
      SUBROUTINE FILEM2(LIST)
      INTEGER TIME(18,4),POLICY(18,4,7,4),PRTY(18,4),CPTY(4),OPER,
     *FLAG,G,T,L,W,COD,DUE,PROC,MACH,PRATE,PROW(26),
     *SIZE,NM,N,TROW,INDX(4,18),MASTER(1500,4),MSTR(2000),
     *LINKPR(1500),LINKSR(1500),LNKPR(2000),LNKSR(2000),ROW,TRNSFR(5),
     *NAR(2),TAIL(72),HEAD(72),LSIZE(72),TAL(1500),HED(1500),
     *LSIZ(1500),BEHIND,AHEAD,ITAIL,POLI,NUM,LIST,INDEX,R,
     *OVRLAP,IDLE,REQTIM,RW,RW1
      COMMON /MODL10/ TIME,POLICY,PRTY,CPTY,OPER,FLAG,G,T,L,W,COD,
     *DUE,PROC,INDX,PRATE,MACH,PROW,TROW,NM
      COMMON /MODL11/ MASTER,MSTR,LINKPR,LINKSR,LNKPR,LNKSR,HEAD,
     *TAIL,HED,TAL,LSIZE,LSIZ,TRNSFR,NAR
C
C   IF THE MASTER STORAGE ARRAY IS FULL, STOP THE PROGRAM
C
      IF(NAR(2).GT.0)GO TO 20
      WRITE(5,10)
10    FORMAT(5X,'MASTER STORAGE ARRAY OVERFLOW')
      STOP
C
C   IF THE LIST VALUE IS IMPROPER, STOP THE PROGRAM
C
20    IF((LIST.GE.1).AND.(LIST.LE.1500))GO TO 40
      WRITE(5,30)
30    FORMAT(5X,I5,' IS AN IMPROPER VALUE FOR FILE LIST')
      STOP
C
C   INCREMENTING THE LIST SIZE COUNTER
C
40    LSIZ(LIST)=LSIZ(LIST)+1
C
C   IF THIS IS NOT THE FIRST RECORD IN THE LIST, CONTINUE
C
      IF(LSIZ(LIST).EQ.1)GO TO 50
C
C   INSERTING BEFORE FIRST RECORD IN LIST
C
      ROW=NAR(2)
      NAR(2)=LNKSR(ROW)
      IF(NAR(2).GT.0) LNKPR(NAR(2))=0
      IHEAD=HED(LIST)
      LNKPR(IHEAD)=ROW
      LNKSR(ROW)=IHEAD
      LNKPR(ROW)=0
      HED(LIST)=ROW
      GO TO 60
C
C   INSERTING FIRST RECORD IN LIST
C
50    ROW=NAR(2)
```

```
      NAR(2)=LNKSR(ROW)
      IF(NAR(2).GT.0) LNKPR(NAR(2))=0
      LNKSR(ROW)=0
      HED(LIST)=ROW
      TAL(LIST)=ROW
C
C   TRANSFERING DATA TO MASTER ARRAY
C
60    MSTR(ROW)=TRNSFR(1)
      RETURN
      END
```

```
C
C     JUNE 5. 1984
C
C     THIS SUBPROGRAM IS THE FIRST OF THREE SUBPROGRAMS THAT RECALCULATES
C     THE DUE DATES AND START TIMES FOR TASKS IN THE MASTER ARRAY.  THIS
C     SUBROUTINE IS ONLY CALLED IF A CONFLICT EXISTS BETWEEN A TASK IN THE
C     MASTER ARRAY AND A TASK BEING ASSIGNED TO THE ARRAY.
C
C     AUTHOR:  ERIC SCHILD
C
      SUBROUTINE PSHBK1(R.OVRLAP)
      INTEGER TIME(18.4),POLICY(18,4.7.4),PRTY(18.4).CPTY(4).OPER.
     *FLAG,G.T.L,W.COD.DUE.PROC,MACH,PRATE,PROW(26),
     *SIZE.NM.N.TROW,INDX(4.18).MASTER(1500.4),MSTR(2000).
     *LINKPR(1500).LINKSR(1500).LNKPR(2000),LNKSR(2000),ROW.TRNSFR(5),
     *NAR(2).TAIL(72).HEAD(72),LSIZE(72).TAL(1500),HED(1500).
     *LSIZ(1500).BEHIND.AHEAD.ITAIL,POLI.NUM.LIST,INDEX,R.
     *OVRLAP.IDLE.REQTIM.RW.RW1
      COMMON /MODL10/ TIME.POLICY,PRTY,CPTY.OPER.FLAG.G.T.L.W,COD.
     *DUE,PROC,INDX,PRATE,MACH,PROW,TROW,NM
      COMMON /MODL11/ MASTER.MSTR,LINKPR,LINKSR.LNKPR.LNKSR.HEAD.
     *TAIL,HED.TAL.LSIZE.LSIZ,TRNSFR,NAR
C
C     IF TASK OVER LAP IS ZERO. RETURN
C
5     IF(OVRLAP.LE.0)GO TO 15
      BEHIND=MASTER(R,2)
C
C     RECALCULATING DUE DATES AND START TIMES
C
      MASTER(R.1)=MASTER(R.1)-OVRLAP
      MASTER(R.2)=MASTER(R.2)-OVRLAP
C
C     CHECKING RAW MATERIAL SOURCE FOR TASK THAT HAS BEEN PUSHED BACK
C
C     IF TASK IS DEVELOPED ON DRYER OPERATION. SKIP CALL TO PUSHBK2
C
      IF(MSTR(HED(R)).LE.0)GO TO 10
      CALL PSHBK2(MASTER(R.2),R)
C
C     CHECKING NEXT TASK IN LIST
C
10    R=LINKSR(R)
C
C     IF LAST ROW CONSIDERED WAS NOT THE TAIL OF THE LIST. CONTINUE
C
      IF(R.EQ.0)GO TO 15
C
C     COMPUTING IDLE TIME
C
      IDLE=BEHIND-MASTER(R,1)
C
C     UPDATING OVERLAP
C
      OVRLAP=OVRLAP-IDLE
      GO TO 5
15    CONTINUE
      RETURN
      END
```

```
C
C     JUNE 5, 1984
C
C     THIS SUBPROGRAM IS THE SECOND OF THREE SUBPROGRAMS THAT RECALCULATES
C     THE DUE DATES AND START TIMES FOR TASKS IN THE MASTER ARRAY.  THIS
C     SUBROUTINE IS CALLED BY PUSHBK1 AFTER A CONFLICT EXISTS BETWEEN THE
C     TASK BEING ASSIGNED AND A TASK ALREADY INCLUDED IN THE MASTER ARRAY.
C     THIS SUBPROGRAM ENSURES THAT THE SOURCES (ON PRECEEDING OPERATIONS)
C     THAT PROVIDE THE RAW MATERIAL FOR THE TASK BEING PUSHED BACK ARE
C     AVAILABLE ON TIME.
C
C     AUTHOR:  ERIC SCHILD
C
      SUBROUTINE PSHBK2(REQTIM,R)
      INTEGER TIME(18,4),POLICY(18,4,7,4),PRTY(18,4),CPTY(4),OPER,
     *FLAG,G,T,L,W,COD,DUE,PROC,MACH,PRATE,PROW(26),
     *SIZE,NM,N,TROW,INDX(4,18),MASTER(1500,4),MSTR(2000),
     *LINKPR(1500),LINKSR(1500),LNKPR(2000),LNKSR(2000),ROW,TRNSFR(5),
     *NAR(2),TAIL(72),HEAD(72),LSIZE(72),TAL(1500),HED(1500),
     *LSIZ(1500),BEHIND,AHEAD,ITAIL,POLI,NUM,LIST,INDEX,R,
     *OVRLAP,IDLE,REQTIM,RW,RW1
      COMMON /MODL10/ TIME,POLICY,PRTY,CPTY,OPER,FLAG,G,T,L,W,COD,
     *DUE,PROC,INDX,PRATE,MACH,PROW,TROW,NM
      COMMON /MODL11/ MASTER,MSTR,LINKPR,LINKSR,LNKPR,LNKSR,HEAD,
     *TAIL,HED,TAL,LSIZE,LSIZ,TRNSFR,NAR
C
C     IF THE LAST ROW CONSIDERED WAS NOT THE TAIL OF LIST, CONTINUE
C
      RW=HED(R)
15    IF(RW.EQ.0)GO TO 35
      RW1=MSTR(RW)
C
C     COMPUTING THE OVERLAP
C
      OVRLAP=MASTER(RW1,1)-REQTIM
C
C     IF NO OVERLAP, CONSIDER THE NEXT RAW MATERIAL SOURCE IN LIST
C
20    IF(OVRLAP.LE.0)GO TO 30
      BEHIND=MASTER(RW1,2)
C
C     RECALCULATING DUE DATES AND START TIMES
C
      MASTER(RW1,1)=MASTER(RW1,1)-OVRLAP
      MASTER(RW1,2)=MASTER(RW1,2)-OVRLAP
C
C     CHECKING RAW MATERIAL SOURCE FOR TASK THAT HAS BEEN PUSHED BACK
C
C     IF TASK IS DEVELOPED ON DRYER OPERATION, SKIP CALL TO PUSHBK3
C
      IF(MSTR(HED(RW1)).LE.0)GO TO 25
      CALL PSHBK3(MASTER(RW1,2),RW1)
C
C     CHECKING NEXT TASK IN LIST
C
25    RW1=LINKSR(RW1)
C
C     IF LAST ROW CONSIDERED WAS NOT THE TAIL OF THE LIST, CONTINUE
C
      IF(RW1.EQ.0)GO TO 30
```

```
C
C   COMPUTING IDLE TIME
C
      IDLE=BEHIND-MASTER(RW1,1)
C
C   UPDATING OVERLAP
C
      OVRLAP=OVRLAP-IDLE
      GO TO 20
30    RW=LNKSR(RW)
      GO TO 15
35    CONTINUE
      RETURN
      END
```

```
C
C     JUNE, 5, 1984
C
C     THIS SUBPROGRAM IS THE THIRD OF THREE SUBPROGRAMS THAT RECALCULATES
C     THE DUE DATES AND START TIMES FOR TASKS IN THE MASTER ARRAY.  THIS
C     SUBROUTINE IS CALLED BY PUSHBK1 AFTER A CONFLICT EXISTS BETWEEN THE
C     TASK BEING ASSIGNED AND A TASK ALREADY INCLUDED IN THE MASTER ARRAY.
C     THIS SUBPROGRAM ENSURES THAT THE SOURCES (ON PRECEEDING OPERATIONS)
C     THAT PROVIDE THE RAW MATERIAL FOR THE TASK BEING PUSHED BACK ARE
C     AVAILABLE ON TIME.
C
C     AUTHOR:  ERIC SCHILD
C
      SUBROUTINE PSHBK3(REQTIM,R)
      INTEGER TIME(18,4),POLICY(18,4,7,4),PRTY(18,4),CPTY(4),OPER,
     *FLAG,G,T,L,W,COD,DUE,PROC,MACH,PRATE,PROW(26),
     *SIZE,NM,N,TROW,INDX(4,18),MASTER(1500,4),MSTR(2000),
     *LINKPR(1500),LINKSR(1500),LNKPR(2000),LNKSR(2000),ROW,TRNSFR(5),
     *NAR(2),TAIL(72),HEAD(72),LSIZE(72),TAL(1500),HED(1500),
     *LSIZ(1500),BEHIND,AHEAD,ITAIL,POLI,NUM,LIST,INDEX,R,
     *OVRLAP,IDLE,REQTIM,RW,RW1
      COMMON /MODL10/ TIME,POLICY,PRTY,CPTY,OPER,FLAG,G,T,L,W,COD,
     *DUE,PROC,INDX,PRATE,MACH,PROW,TROW,NM
      COMMON /MODL11/ MASTER,MSTR,LINKPR,LINKSR,LNKPR,LNKSR,HEAD,
     *TAIL,HED,TAL,LSIZE,LSIZ,TRNSFR,NAR
C
C     IF THE LAST ROW CONSIDERED WAS NOT THE TAIL OF LIST, CONTINUE
C
      RW=HED(R)
15    IF(RW.EQ.0)GO TO 40
      RW1=MSTR(RW)
C
C     COMPUTING THE OVERLAP
C
      OVRLAP=MASTER(RW1,1)-REQTIM
C
C     IF NO OVERLAP, CONSIDER THE NEXT RAW MATERIAL SOURCE IN LIST
C
20    IF(OVRLAP.LE.0)GO TO 35
      BEHIND=MASTER(RW1,2)
C
C     RECALCULATING DUE DATES AND START TIMES
C
      MASTER(RW1,1)=MASTER(RW1,1)-OVRLAP
      MASTER(RW1,2)=MASTER(RW1,2)-OVRLAP
C
C     CHECKING RAW MATERIAL SOURCE FOR TASK THAT HAS BEEN PUSHED BACK
C
C     IF TASK IS DEVELOPED ON DRYER OPERATION, WRITE ERROR
C
      IF(MSTR(HED(RW1)).LE.0)GO TO 30
      WRITE(5,25)
25    FORMAT(5X,'ERROR IN LINKED LIST--TRACES TOO MANY OPERATIONS')
      WRITE(5,26) MSTR(HED(RW1))
26    FORMAT(T2,'ERROR IN LINKED LIST-',I5)
      STOP
C
C     CHECKING NEXT TASK IN LIST
C
30    RW1=LINKSR(RW1)
```

```
C
C    IF LAST ROW CONSIDERED WAS NOT THE TAIL OF THE LIST, CONTINUE
C
      IF(RW1.EQ.0)GO TO 35
C
C    COMPUTING IDLE TIME
C
      IDLE=BEHIND-MASTER(RW1,1)
C
C    UPDATING OVERLAP
C
      OVRLAP=OVRLAP-IDLE
      GO TO 20
35    RW=LNKSR(RW)
      GO TO 15
40    CONTINUE
      RETURN
      END
```

A.2.3 Variable Definitions and Listing of Time-phasing Program

CODE - Veneer code.

COD1(i,j,k) - Veneer task code array; used to record the veneer codes of the ith task assigned to the jth machine of the kth operation. i=1,50, j=1,18, k=1,3.

COMPL - Completion time of veneer task.

COUNT(i,j) - Total number of tasks assigned to ith machine of the jth operation. i=1,18, j=1,3.

CPTY(i) - Machine capacity of ith operation. i=1,3.

DAY - Current day.

DDAT(i,j,k) - Due date of the ith task assigned to the jth machine of the kth operation. i=1,50, j=1,18, k=1,3.

END(i) - Completion time of ith operation. i=1,3.

GMIN - Earliest start time among all operations.

NDAYS(i) - Makespan of ith operation. i=1,3.

NTSK(i,j,k) - Total number of tasks assigned to the kth machine during the ith shift of the jth day. i=1,3, j=1,100, k=1,18.

NXT(i) - Completion time of shift i. i=1,3.

OP - Operation with the earliest start time.

PR(i,j,k) - Production rate of the ith task assigned to the jth machine of the kth operation. i=1,50, j=1,18, k=1,3.

PRAT - Production rate of veneer task.

PRID - Veneer code identification.

PRTM(i,j,k) - Processing time required for the ith task assigned to the jth machine of the kth operation. i=1,50, j=1,18, k=1,3.

SDAT(i,j,k) - Start time of the ith task assigned to the jth machine of the kth operation. i=1,50, j=1,18, k=1,3.

SHF - Current shift.

SHFT(i) - Starting shift for ith operation. i=1,4.

START(i) - Starting day for ith operation. i=1,4.

STRT - Start time of veneer task.

TSK- Task number.

```
C
C   FEB. 22. 1984
C
C   THIS PROGRAM IS THE MAIN DRIVER FOR A ROUTINE THAT TIME PHASES THE
C   THREE SCHEDULED LISTS (CREATED IN SUBROUTINE TALLY).  THIS PROGRAM
C   SPECIFICALLY READS IN EACH SCHEDULED LIST AND CALLS THE APPROPRIATE
C   SUBROUTINES TO TIME PHASE THE TASKS.
C
C   AUTHOR:  ERIC SCHILD
C
      PROGRAM TSKALC
      INTEGER CPTY(3),OPER,MACH,SHF,DAY,NTSK(3,100,18),TSK,PFLAG,
     *PRID,CODI(50,18,3),COUNT(18,3),OP,CODE,COMPL,STRT,NDAYS(3),
     *PRAT,SHFT(4),GMIN,END(3),START(4),PR(50,18,3),DDAT(50,18,3),
     *SDAT(50,18,3),PRTM(50,18,3),NXT(3)
      COMMON /MODL14/ END,START,OP,GMIN,NDAYS,OPER,CPTY,COUNT,NTSK,
     *MACH,PR,CODI,DDAT,SDAT,PRTM,SHFT
C
C   MACHINE CAPACITY FOR THE THREE OPERATIONS
C
      DATA CPTY/18,5,7/
      GMIN=30000
C
      DO 55 OPER=1,3
C
C   INITIALIZING TASK COUNTER
C
      DO 8 J=1,18
      COUNT(J,OPER)=0
8     CONTINUE
C
      GO TO (10,15,20) OPER
C
C   PATCHER OPERATION
C
10    CALL ASSIGN(1,'SY:PATTSK.DAT')
      GO TO 40
C
C   JOINTER OPERATION
C
15    CALL ASSIGN(1,'SY:JOITSK.DAT')
      GO TO 40
C
C   DRYER OPERATION
C
20    CALL ASSIGN(1,'SY:DRYTSK.DAT')
C
C   READING IN THE TASKS FROM THE CURRENT SCHEDULED LIST.
C
40    READ(1,45,END=50) COMPL,STRT,CODE,PRAT,MACH
45    FORMAT(I2,2I6,I4,I6,I3)
      COUNT(MACH,OPER)=COUNT(MACH,OPER)+1
      PRTM(COUNT(MACH,OPER),MACH,OPER)=COMPL-STRT
      CODI(COUNT(MACH,OPER),MACH,OPER)=CODE
      DDAT(COUNT(MACH,OPER),MACH,OPER)=COMPL-21600
      SDAT(COUNT(MACH,OPER),MACH,OPER)=STRT-21600
      PR(COUNT(MACH,OPER),MACH,OPER)=PRAT
      GO TO 40
50    CONTINUE
      CALL CLOSE(1)
```

```
55    CONTINUE
C
      CALL DAYS
      CALL STIME
      CALL TSKFL2
      STOP
      END
```

```
C
C     FEB. 22, 1984
C
C     THIS SUBROUTINE COMPUTES THE TOTAL NUMBER OF DAYS THAT ARE RE-
C     QUIRED TO PRODUCE ALL OF THE TASKS ON THE M MACHINES OF THE
C     CURRENT OPERATION.  IT ALSO DETERMINES THE EARLIEST START TIME
C     AND LATEST COMPLETION TIME FOR THE CURRENT OPERATION.
C
C
C     AUTHOR:  ERIC SCHILD
C
      SUBROUTINE DAYS
      INTEGER CPTY(3).OPER.MACH.SHF.DAY,NTSK(3.100.18).TSK.PFLAG.
     *PRID.COD1(50.18.3).COUNT(18.3).OP.CODE.COMPL.STRT.NDAYS(3).
     *PRAT.SHFT(4).GMIN.END(3).START(4).PR(50.18.3).DDAT(50.18.3).
     *SDAT(50.18.3).PRTM(50.18.3).NXT(3)
      COMMON /MODL14/ END.START.OP.GMIN.NDAYS.OPER.CPTY.COUNT.NTSK.
     *MACH.PR.COD1.DDAT,SDAT.PRTM,SHFT
C
C     INITIALIZING START TIME AND COMPLETION TIME STATISTICS
C
      DO 50 OPER=1.3
      START(OPER)=30000
      END(OPER)=-30000
      DO 20 MACH=1.CPTY(OPER)
C
C     START TIME COMPARISON
C
      IF(SDAT(COUNT(MACH.OPER).MACH.OPER).GE.START(OPER))GO TO 10
      START(OPER)=SDAT(COUNT(MACH.OPER),MACH.OPER)
C
C     COMPLETION TIME COMPARISON
C
10    IF(DDAT(1.MACH.OPER).LE.END(OPER))GO TO 20
      END(OPER)=DDAT(1.MACH.OPER)
20    CONTINUE
C
C     RESCALING START TIMES FOR EACH TASK ASSIGNED TO CURRENT OPERATION
C
      DO 40 MACH=1.CPTY(OPER)
      DO 30 TSK=1.COUNT(MACH.OPER)
      SDAT(TSK.MACH.OPER)=SDAT(TSK.MACH.OPER)-START(OPER)+1440
30    CONTINUE
40    CONTINUE
C
C     COMPUTING MAKESPAN
C
      NDAYS(OPER)=NINT(FLOAT(END(OPER)-START(OPER))/1440.)
C
C     IF START TIME IS SMALLER THAN EARLIEST START TIME AMONG THE
C     OPERATIONS ALREADY CHECKED. RETAIN START TIME.
C
      IF(START(OPER).GE.GMIN)GO TO 50
      GMIN=START(OPER)
      OP=OPER
50    CONTINUE
      RETURN
      END
```

```
C
C     APRIL 8,1984
C
C     THIS SUBPROGRAM IS USED TO WRITE THE START TIMES FOR EACH
C     OPERATION IN A FILE NAMED START.DAT.  THIS IS DONE SO THAT THE
C     CORRECT STARTING TIMES WILL BE USED FOR EACH OPERATION IN THE
C     REPORT GENERATION PHASE.
C
C     AUTHOR:  ERIC SCHILD
C
      SUBROUTINE STIME
      INTEGER CPTY(3),OPER,MACH,SHF,DAY,NTSK(3,100,18),TSK,PFLAG,
     *PRID,COD1(50,18,3),COUNT(18,3),OP,CODE,COMPL,STRT,NDAYS(3),
     *PRAT,SHFT(4),GMIN,END(3),START(4),PR(50,18,3),DDAT(50,18,3),
     *SDAT(50,18,3),STD(4),B,PRTM(50,18,3),NXT(3)
      REAL A,C,DIFF
      COMMON /MODL14/ END,START,OP,GMIN,NDAYS,OPER,CPTY,COUNT,NTSK,
     *MACH,PR,COD1,DDAT,SDAT,PRTM,SHFT
C
      CALL ASSIGN(3,'SY:START.DAT')
      DIFF=ABS(FLOAT(GMIN))/1440.
      STD(4)=NINT(DIFF)+1
      SHFT(4)=1
      DO 20 I=1,3
C
C     IF SMALLEST START TIME, SET START TIME EQUAL TO 1.
C
      IF(I.NE.OP) GO TO 5
      STD(I)=1
      SHFT(I)=1
      GO TO 20
C
C     CALCULATING START TIME
C
5     A=FLOAT(START(I)-GMIN)/1440.
      B=IFIX(A)
      C=A-FLOAT(B)
      STD(I)=B+1
      IF(C.GE.1./3.)GO TO 10
      SHFT(I)=1
      GO TO 20
10    IF(C.GE.2./3.)GO TO 15
      SHFT(I)=2
      GO TO 20
15    SHFT(I)=3
20    CONTINUE
      WRITE(3,25) DIFF,STD(4),SHFT(4),(STD(M),SHFT(M),M=1,3)
25    FORMAT(I2,F6.2,8I2)
      RETURN
      END
```

```
C
C     JUNE 5, 1984
C
C     THIS SUBPROGRAM TIME PHASES THE SCHEDULED TASKS ASSIGNED TO EACH
C     MACHINE OF THE CURRENT OPERATION.
C
C     AUTHOR:  ERIC SCHILD
C
        SUBROUTINE TSKFL2
        INTEGER CPTY(3),OPER,MACH,SHF,DAY,NTSK(3,100,18),TSK,PFLAG,
     *PRID,COD1(50,18,3),COUNT(18,3),OP,CODE,COMPL,STRT,NDAYS(3),
     *PRAT,SHFT(4),GMIN,END(3),START(4),PR(50,18,3),DDAT(50,18,3),
     *SDAT(50,18,3),NSHFT,TIME,TIMAV,PRTM(50,18,3),NXT(3)
        COMMON /MODL14/ END,START,OP,GMIN,NDAYS,OPER,CPTY,COUNT,NTSK,
     *MACH,PR,COD1,DDAT,SDAT,PRTM,SHFT
C
C     SETTING THE SHIFT CHANGE ARRAY
C
        NXT(1)=480
        NXT(2)=960
        NXT(3)=1440
C
C     IF ALL OPERATIONS HAVE BEEN TIME PHASED, RETURN
C
        DO 90 OPER=1,3
C
        GO TO (1,2,3) OPER
C
C     PATCHER OPERATION
C
1       CALL ASSIGN(4,'SY:SEQ2.DAT')
        GO TO 4
C
C         JOINTER OPERATION
C
2       CALL ASSIGN(4,'SY:SEQ3.DAT')
        GO TO 4
C
C     DRYER OPERATION
C
3       CALL ASSIGN(4,'SY:SEQ4.DAT')
C
C     IF ALL MACHINES HAVE BEEN TIME PHASED, TIME PHASE NEXT OPERATION
C
4       DO 80 MACH=1,CPTY(OPER)
        NSHFT=SHFT(OPER)
C
C     INITIALIZING FORMATTING ARRAY
C
        DO 10 K=1,NDAYS(OPER)
        DO 5 J=1,3
        NTSK(J,K,MACH)=0
5       CONTINUE
10      CONTINUE
C
C     INITIALIZING TASK COUNTER
C
        TSK=COUNT(MACH,OPER)
        PFLAG=0
        DO 70 DAY=1,NDAYS(OPER)
```

```
      IF(DAY.EQ.1)GO TO 15
      NSHFT=1
15    DO 60 SHF=NSHFT,3
C
C    SET CURRENT TIME TO END OF SHIFT
C
      TIME=DAY*1440+NXT(SHF)
C
C    IF TASK COUNTER LESS THAN 1, CONTINUE
C
20    IF(TSK.LT.1) GO TO 50
C
C    IF START TIME FOR TASK IS LESS THAN CURRENT TIME, ASSIGN TASK
C    TO CURRENT SHIFT
C
      IF(TIME.GT.SDAT(TSK,MACH,OPER))GO TO 30
      IF(PFLAG.EQ.1)GO TO 55
C
C    CREATING IDLE SHIFT
C
      PRID=-1
      PQTY=0
      WRITE(4,85) PRID,PQTY,MACH,SHF
      GO TO 60
C
C    COMPUTING AVAILABLE TIME
C
30    TIMAV=TIME-SDAT(TSK,MACH,OPER)
      PFLAG=0
C
C    IF PROCESSING TIME FOR TASK GREATER THATN AVAILABLE TIME, ASSIGN
C    PART OF TASK TO CURRENT SHIFT
C
      IF(PRTM(TSK,MACH,OPER).GT.TIMAV)GO TO 40
C
C    ASSIGNING TASK TO CURRENT SHIFT
C
      PRID=COD1(TSK,MACH,OPER)
      PQTY=FLOAT(PRTM(TSK,MACH,OPER))*FLOAT(PR(TSK,MACH,OPER))/100000.
      PRTM(TSK,MACH,OPER)=0
      NTSK(SHF,DAY,MACH)=NTSK(SHF,DAY,MACH)+1
      WRITE(4,85) PRID,PQTY,MACH,SHF,DDAT(TSK,MACH,OPER)
C
C    DEINCREMENTING TASK COUNTER
C
      TSK=TSK-1
      PFLAG=1
      GO TO 20
C
C    ASSIGNING PART OF TASK TO CURRENT SHIFT
C
40    PRID=COD1(TSK,MACH,OPER)
      PQTY=FLOAT(TIMAV)*FLOAT(PR(TSK,MACH,OPER))/100000.
      PRTM(TSK,MACH,OPER)=PRTM(TSK,MACH,OPER)-TIMAV
      NTSK(SHF,DAY,MACH)=NTSK(SHF,DAY,MACH)+1
      WRITE(4,85) PRID,PQTY,MACH,SHF,DDAT(TSK,MACH,OPER)
      SDAT(TSK,MACH,OPER)=TIME
      GO TO 60
50    IF(PFLAG.EQ.1)GO TO 55
C
```

```
C     CREATING IDLE SHIFT
C
        PRID=-1
        PQTY=0
        WRITE(4,85) PRID,PQTY,MACH,SHF
55      PFLAG=0
60      CONTINUE
70      CONTINUE
80      CONTINUE
        CALL CLOSE(4)
85      FORMAT(I4,F6.2,I3,I4,I6)
        CALL WRNTSK
90      CONTINUE
        RETURN
        END
```

```
C
C    FEB. 22. 1984
C
C    THIS SUBPROGRAM IS USED TO CONSTRUCT THE FORMATTING FILES
C    (ENTRY2.DAT,ENTRY3.DAT.AND ENTRY4.DAT) THAT WILL BE USED ALONG
C    SIDE THE SEQUENCING FILES (SEQ2.DAT.SEQ3.DAT. AND SEQ4.DAT) IN
C    THE REPORT GENERATION PHASE.
C
C    AUTHOR:  ERIC SCHILD
C
      SUBROUTINE WRNTSK
      INTEGER CPTY(3).OPER.MACH.SHF.DAY.NTSK(3.100.18).TSK.PFLAG.
     *PRID.COD1(50.18.3).COUNT(18.3).OP.CODE.COMPL.STRT.NDAYS(3).
     *PRAT.SHFT(4),GMIN.END(3).START(4).PR(50.18.3).DDAT(50.18.3).
     *SDAT(50,18.3),MAXTK(3,50),PRTM(50,18.3),NXT(3)
      COMMON /MODL14/ END.START.OP.GMIN.NDAYS.OPER.CPTY.COUNT.NTSK.
     *MACH.PR.COD1.DDAT,SDAT,PRTM,SHFT
      DO 4 J=1,NDAYS(OPER)
      DO 3 K=1,3
      MAXTK(K,J)=1
3     CONTINUE
4     CONTINUE
      GO TO (5.10.15) OPER
C
C    PATCHER OPERATION
C
5     CALL ASSIGN(1,'SY:ENTRY2.DAT')
      GO TO 20
C
C    JOINTER OPERATION
C
10    CALL ASSIGN(1,'SY:ENTRY3.DAT')
      GO TO 20
C
C    DRYER OPERATION
C
15    CALL ASSIGN(1,'SY:ENTRY4.DAT')
20    DO 35 DAY=1,NDAYS(OPER)
      DO 30 SHF=1,3
      DO 25 MACH=1,CPTY(OPER)
C
C    IF TASK COUNTER FOR A GIVEN MACHINE ON A GIVEN SHIFT IS LESS
C    THAN THE MAXIMUM NUMBER OF TASKS ALREADY ASSIGNED DURING THIS
C    SHIFT, SKIP
C
      IF(NTSK(SHF.DAY.MACH).LT.MAXTK(SHF.DAY)) GO TO 25
      MAXTK(SHF.DAY)=NTSK(SHF.DAY.MACH)
25    CONTINUE
30    CONTINUE
35    CONTINUE
C
C    WRITING FORMATTING VARIABLES ON THE APPROPRIATE FILE
C
      DO 45 DAY=1,NDAYS(OPER)
      WRITE(1,40) (MAXTK(SHF.DAY),SHF=1,3)
40    FORMAT(I2,3I2)
45    CONTINUE
      CALL CLOSE(1)
      RETURN
      END
```

APPENDIX A.3 VARIABLE DEFINITIONS AND PROGRAM LISTING FOR PHASE THREE

CONST - Panel construction.

CPTY(i) - Machine capacity for the ith operation. i=1,4.

DASH(i) - A character array of dashes. i=1,132.

DAY - Current day..

DAYS - Day a task is due.

DDATE - Task due date.

DIFF - The time difference between the start of the first operation and the start of the spreader operation.

DNAM(i) - A character array giving the day header, "DAY:'. i=1,5.

EQUAL(i) - A character array of equal signs. i=1,132.

EXT(i) - A character array indicating whether the panel is an exterior or an interior type panel. i=1,3.

GRADE(i) - A character array giving the panel grade. i=1,5.

HEAD(i,j) - A character array giving the column headings for the ith operation. i=1,4, j=1,33.

J - Operation index presently being formatted.

LEN - Panel or veneer length.

LIST(i) - Operations to be formatted. i=1,4.

MACHID - Machine to be used to process a task.

MAXTK(i,j) - Maximum number of tasks produced during the ith shift on the jth day. i=1,3, j=1,50.

N - The machine that the last task was formatted on.

OPER(i) - A character array giving the name of each operation, "SPREADER", PATCHER", "JOINTER", or "DRYER". i=1,9.

PERIOD - The shift that a task will be processed.

PERNAM(i,j) - A character array giving the name of the ith shift, "GYD", "DAYS", or "SWG". i=1,4, j=1,6.

PMACH(i) - Number of tasks assigned to the ith machine on the current shift. i=1,18.

PNUM(i) - Total number of lines that have been used to format the ith machine. i=1,18.

PQTY - Production quantity.

PRODID - Task identification code.

PRODQT - Production quantity.

S - The shift a task is due.

SHF(i,j) - A character array used to translate S into its appropriate character string (GYD, DAYS, or SWG). i=1,3, j=1,4.

SHFT(i) - Starting shift for ith operation. i=1,4.

SHIFT - Current shift.

SNAM(i) - A character array giving the shift header, "SHIFT:". i=1,7.

START(i) - Starting day for ith operation.

THK(i) - A character arrray giving the panel thickness. i=1,5.

TYPE(i) - A character array giving the panel type (sanded, siding, duraply, marine, or sheathing). i=1,3.

WIDTH(i) - A character array giving the veneer width. i=1,2.

```
C
C  DATE: 25 AUG. 1983
C
C  THIS PROGRAM INITIATES THE REPORT GENERATION PHASE.
C
C  AUTHOR: ERIC SCHILD
C
      PROGRAM OUTPT
      INTEGER MAXTK(3.50).CONST.LEN.CPTY(4).LFLAG.PRODID.SHIFT.MACHID.
     *DAY.J,N.PNUM(18).LIST(4).PQTY.P,MACH(18).PERIOD.PMACH(18).
     *START(4).DDATE.SHFT(4).DAYS.S
      BYTE THK(8).GRADE(8).TYPE(5).EXT(5).WIDTH(2).PERNAM(4.6).
     *DAT(9).HEAD(4.33).DASH(132).DNAM(5).SNAM(7).
     *OPER(4.8).EQUAL(132).SHF(3.4)
      REAL DIFF,PRODQT
      COMMON /MODEL1/ MAXTK.CPTY.CONST.LEN.START.DAYS.DDATE.S.SHFT
      COMMON /MODEL2/ PRODQT.PRODID.SHIFT.MACHID.N.PNUM.DIFF.
     *PMACH.J.DAY.PERIOD.LFLAG.PQTY.SHF
      COMMON /MODEL3/ THK.GRADE.TYPE.EXT.WIDTH
      COMMON /MODEL4/ PERNAM.HEAD.DASH.DNAM.SNAM.OPER.EQUAL
C
C  READS THE NECESSARY CHARACTER STRINGS IN PER.DAT TO FACILITATE
C  THE CONTRUCTION OF THE SHIFT HEADERS. OPERATION HEADERS.
C  AND MACHINE HEADERS.
C
      CALL ASSIGN(3.'SY:PER.DAT')
      DO 10 L=1.4
      READ (3.5) (OPER(L.M),M=1.8).(PERNAM(L.M),M=1.6).
     Z(HEAD(L,M).M=1.33)
 5    FORMAT (8A1.6A1.33A1)
 10   CONTINUE
      READ (3.15) (DNAM(M).M=1.4).(SNAM(M).M=1.6)
 15   FORMAT (4A1.6A1)
      CALL CLOSE(3)
C
C   READING IN THE START TIMES FOR EACH OPERATION
C
      CALL ASSIGN(3.'SY:START.DAT')
 20   READ(3.25) DIFF.(START(M).SHFT(M),M=1.4)
 25   FORMAT(T2.F6.2.8I2)
      CALL CLOSE(3)
C
C  SETTING MACHINE CAPACITY FOR THE FOUR OPERATIONS.
C
      DATA CPTY/6.18.5.7/
      DATA SHF/'G'.'D'.'S','Y'.'A'.'W'.'D'.'Y'.'G'.' ','S'.' '/
C
      DO 35 L=1,132
      DASH(L)='-'
      EQUAL(L)='='
 35   CONTINUE
C
      CALL DATE(DAT)
C
C  MAIN HEADER.
C
      WRITE (6.40) (DAT(M).M=1.9).(DASH(M),M=1.131).
     *(DASH(M).M=1.131)
 40   FORMAT('1'.'DRY-END SCHEDULE'/T2.'CBP-LEBANON'.95X.
     Z9A1//T2.131A1/T2.131A1)
```

```
C
C
      WRITE(5,45)
 45   FORMAT(T2,'WHICH OPERATION SCHEDULES'/
     *T2,'WOULD YOU LIKE LISTED: ',$)
      ACCEPT*,(LIST(M),M=1,4)
      J=1
 50   IF (J.GT.4) GO TO 75
      DO 55 K=1,4
      IF(LIST(K).EQ.J)GO TO 60
 55   CONTINUE
      GO TO 70
C
C   OPERATION HEADER.
C
 60   WRITE (6,65) (OPER(J,M),M=1,8)
 65   FORMAT (T2,8A1/T2,'OPERATION')
      CALL OPENFL
      CALL TCODE
 70   J=J+1
      GO TO 50
 75   CONTINUE
      STOP
      END
```

```
C
C
C     FEB. 22, 1984
C
C     THE PRIMARY PURPOSE OF THIS SUBPROGRAM IS TO OPEN AND READ THE
C     ENTRY[J.DAT FILES.  IT ALSO IS USED TO OPEN THE SCHEDULED TASK
C     FILES (SEQ[J.DAT). OUTPUT FILES (MACH[J.DAT). AND THE ARCHIVAL
C     FILES THAT WILL BE USED TO TRANSLATE THE NUMERIC IDENTIFICATION
C     CODE IN TO READABLE CHARACTER STRINGS.
C
C     AUTHOR:  ERIC SCHILD
C
      SUBROUTINE OPENFL
      INTEGER MAXTK(3,50),CONST,LEN,CPTY(4),LFLAG,PRODID,SHIFT,MACHID,
     *DAY,J,N,PNUM(18),SHFT(4),PQTY,P,MACH(18),PERIOD,PMACH(18),
     *START(4),DDATE,DAYS,S
      BYTE THK(8),GRADE(8),TYPE(5),EXT(5),WIDTH(2),PERNAM(4,6),
     *TIM(8),DAT(9),HEAD(4,33),DASH(132),DNAM(5),SNAM(7),
     *OPER(4,8),EQUAL(132),SHF(3,4)
      REAL DIFF,PRODQT
      COMMON /MODEL1/ MAXTK,CPTY,CONST,LEN,START,DAYS,DDATE,S,SHFT
      COMMON /MODEL2/ PRODQT,PRODID,SHIFT,MACHID,N,PNUM,DIFF,
     *PMACH,J,DAY,PERIOD,LFLAG,PQTY,SHF
      COMMON /MODEL3/ THK,GRADE,TYPE,EXT,WIDTH
      COMMON /MODEL4/ PERNAM,HEAD,DASH,DNAM,SNAM,OPER,EQUAL
      GO TO (10,20,30,40) J
C
C     FILES REQUIRED FOR THE SPREADER OPERATION
C
10    CALL ASSIGN(1,'SY:ENTRY1.DAT')
      CALL ASSIGN(3,'SY:SEQ1.DAT')
      OPEN(UNIT=4,TYPE='OLD',ACCESS='DIRECT',NAME='SY:PANCO.DAT',
     *RECORDSIZE=20,READONLY)
      GO TO 50
C
C     FILES REQUIRED FOR THE PATCHER OPERATION
C
20    CALL ASSIGN(1,'SY:ENTRY2.DAT')
      CALL ASSIGN(3,'SY:SEQ2.DAT')
      OPEN(UNIT=4,TYPE='OLD',ACCESS='DIRECT',NAME='SY:INVDEV.DAT',
     *RECORDSIZE=5,READONLY)
      GO TO 50
C
C     FILES REQUIRED FOR THE JOINTER OPERATION
C
30    CALL ASSIGN(1,'SY:ENTRY3.DAT')
      CALL ASSIGN(3,'SY:SEQ3.DAT')
      OPEN(UNIT=4,TYPE='OLD',ACCESS='DIRECT',NAME='SY:INVDEV.DAT',
     *RECORDSIZE=5,READONLY)
      GO TO 50
C
C     FILES REQUIRED FOR THE DRYER OPERATION
C
40    CALL ASSIGN(1,'SY:ENTRY4.DAT')
      CALL ASSIGN(3,'SY:SEQ4.DAT')
      OPEN(UNIT=4,TYPE='OLD',ACCESS='DIRECT',NAME='SY:DRYDEV.DAT',
     *RECORDSIZE=15,READONLY)
50    CALL ASSIGN(7,'SY:MACH15.DAT')
      CALL ASSIGN(8,'SY:MACH26.DAT')
      CALL ASSIGN(9,'SY:MACH37.DAT')
```

```
      CALL ASSIGN(10,'SY:MACH48.DAT')
C
C   SETTING UP THE FORMATTING VARIABLES
C
      DAY=START(J)
55    READ(1,60,END=70) (MAXTK(SHIFT,DAY),SHIFT=1,3)
60    FORMAT(I2,3I2)
      DAY=DAY+1
      GO TO 55
70    CONTINUE
      CALL CLOSE(1)
      RETURN
      END
```

```
C
C  DATE:  25 AUG. 1983
C
C  THIS SUBPROGRAM READS THE SCHEDULED TASKS FROM SEQ[].DAT.
C  TRANSLATES THE NUMERIC PRODUCT IDENTIFICATION, AND
C  ENSURES THAT THE TASKS ARE ENCODED IN SUB. CHCODE.  IT ALSO
C  CHECKS FOR SHIFT CHANGES AND CALLS THE APPROPRIATE SUBROUTINE
C  WHEN A CHANGE HAS OCCURRED.  AFTER ALL TASKS HAVE BEEN
C  ENCODED. IT MAKES A FINAL CALL TO SUB. RESLT WHERE THE
C  RESULTS ARE SENT TO THE LINE PRINTER.
C
C  AUTHOR: ERIC SCHILD
C
      SUBROUTINE TCODE
C
      INTEGER MAXTK(3.50).CONST.LEN.CPTY(4).LFLAG.PRODID.SHIFT.MACHID.
     *DAY.J.N.PNUM(18).SHFT(4).PQTY.P.MACH(18).PERIOD,PMACH(18).
     *START(4).DDATE.DAYS.S.LPRID
      BYTE THK(8).GRADE(8).TYPE(5).EXT(5).WIDTH(2).PERNAM(4.6).
     *TIM(8).DAT(9).HEAD(4.33).DASH(132).DNAM(5).SNAM(7),
     *OPER(4.8).EQUAL(132).SHF(3.4)
      REAL DIFF.PRODQT
      COMMON /MODEL1/ MAXTK.CPTY.CONST.LEN.START.DAYS.DDATE.S.SHFT
      COMMON /MODEL2/ PRODQT.PRODID.SHIFT.MACHID.N.PNUM.DIFF.
     *PMACH.J.DAY.PERIOD.LFLAG.PQTY.SHF
      COMMON /MODEL3/ THK.GRADE.TYPE.EXT.WIDTH
      COMMON /MODEL4/ PERNAM.HEAD.DASH.DNAM.SNAM.OPER.EQUAL
C
C  INITIALIZING STARTING CONDITIONS
C
      N=1
      DO 5 I=1.18
      PNUM(I)=0
      PMACH(I)=0
 5    CONTINUE
      LFLAG=0
      DAY=START(J)
      SHIFT=SHFT(J)
C
C   READING IN THE SCHEDULED TASK
C
      IF(J.GT.1)GO TO 20
 10   READ (3.15.END=85) PRODID.PQTY.MACHID.PERIOD
 15   FORMAT (I4,I10,I3,I4)
      GO TO 30
 20   READ (3.25.END=85) PRODID.PRODQT.MACHID.PERIOD.DDATE
 25   FORMAT (I4.F6.2.I3.I4.I6)
C
C  INCREMENTING MACHINE COUNTER AND SHIFT COUNTER
C
      CALL DUEDAT
 30   PMACH(MACHID)=PMACH(MACHID)+1
      PNUM(MACHID)=PNUM(MACHID)+1
C
C  AVOID CALL TO SUB. INPT IF PRODID IS NEGATIVE.
C
      IF (PRODID.LT.0) GO TO 38
      IF (PRODID.EQ.LPRID) GO TO 36
      CALL INPT
 36   LPRID=PRODID
```

```
C
C  CHECKING FOR A MACHINE CHANGE
C
 38    IF (MACHID.NE.N)GO TO 40
C
C  MACHINE CHANGE HAS NOT OCCURRED-CHECKING FOR A SHIFT CHANGE
C
       IF (PERIOD.EQ.SHIFT.AND.PRODID.GE.0) GO TO 70
C
C  SHIFT CHANGE HAS OCCURRED
C
       PMACH(N)=PMACH(N)-1
       GO TO 45
C
C  MACHINE CHANGE HAS OCCURRED-SETTING HEADER FLAG SO THAT SHIFT HEADER
C  WILL NOT BE PRINTED IN SUB. HEADER
C
 40    PNUM(N)=PNUM(N)+1
       LFLAG=1
C
C  CHECKING TO SEE IF BLANKS ARE REQUIRED BETWEEN LAST TASK
C  AND THE NEW SHIFT HEADER.
C
  45   IF (PMACH(N).LT.MAXTK(SHIFT,DAY)) GO TO 50
       CALL HEADR
       GO TO 60
  50   CALL BLANKS
       CALL HEADR
C
C  CHECKING FOR IDLE SHIFT FLAG.  IF TRIPPED, SKIP THE CALL TO
C  SUB. CHCODE AND INITIALIZE THE PMACH COUNTER TO ZERO.
C
 60    IF (PRODID.LT.0) GO TO 65
       CALL CHCODE
C
C  RETAIN MACHINE IDENTITY OF PREVIOUSLY ENCODED TASK.
C
       N=MACHID
       GO TO (10,20,20,20) J
 65    N=MACHID
       PMACH(N)=0
       PNUM(N)=PNUM(N)-1
       GO TO (10,20,20,20) J
C
C
 70    N=MACHID
       CALL CHCODE
       GO TO (10,20,20,20) J
 85    CONTINUE
       CALL CLOSE(3)
       CALL CLOSE(7)
       CALL CLOSE(8)
       CALL CLOSE(9)
       CALL CLOSE(10)
       CLOSE(UNIT=4)
C
C  WRITE CONTENTS OF OUTPUT FILES BEFORE RETURNING
C  TO SUB. OUTPT.
C
       CALL RESLT
```

```
.RETURN
 END
```

```
C
C    APRIL 9,1984
C
C    THIS SUBPROGRAM RECALCULATES THE DUE DATES FOR THE TASKS
C    ASSIGNED TO THE PATCHER, JOINTER, AND DRYER OPERATIONS.
C
C    AUTHOR:  ERIC SCHILD
C
      SUBROUTINE DUEDAT
      INTEGER MAXTK(3,50),CONST,LEN,CPTY(4),LFLAG,PRODID,SHIFT,MACHID,
     *DAY,J,N,PNUM(18),SHFT(4),PQTY,P,MACH(18),PERIOD,PMACH(18),
     *START(4),DDATE,DAYS,S
      BYTE THK(8),GRADE(8),TYPE(5),EXT(5),WIDTH(2),PERNAM(4,6),
     *TIM(8),DAT(9),HEAD(4,33),DASH(132),DNAM(5),SNAM(7),
     *OPER(4,8),EQUAL(132),SHF(3,4)
      REAL DIFF,PRODQT
      COMMON /MODEL1/ MAXTK,CPTY,CONST,LEN,START,DAYS,DDATE,S,SHFT
      COMMON /MODEL2/ PRODQT,PRODID,SHIFT,MACHID,N,PNUM,DIFF,
     *PMACH,J,DAY,PERIOD,LFLAG,PQTY,SHF
      COMMON /MODEL3/ THK,GRADE,TYPE,EXT,WIDTH
      COMMON /MODEL4/ PERNAM,HEAD,DASH,DNAM,SNAM,OPER,EQUAL
C
      A=FLOAT(DDATE)/1440.+DIFF+1.
      B=IFIX(A)
      C=A-FLOAT(B)
      DAYS=B
C
      IF(C.GE.1./3.)GO TO 10
      S=1
      RETURN
C
10    IF(C.GE.2./3.)GO TO 20
      S=2
      RETURN
C
20    S=3
      RETURN
      END
```

```
C
C  DATE: 25 AUG. 1983
C
C  THIS SUBPROGRAM USES THE THREE ARCHIVAL FILES OPENED IN SUB OPENFL
C  TO TRANSLATE THE NUMERIC IDENTIFICATION CODE TO A MORE UNDER-
C  STANDABLE FORM.
C
C  AUTHOR:  ERIC SCHILD
C
      SUBROUTINE INPT
      INTEGER MAXTK(3,50),CONST,LEN,CPTY(4),LFLAG,PRODID,SHIFT,MACHID,
     *DAY,J,N,PNUM(18),SHFT(4),PQTY,P,MACH(18),PERIOD,PMACH(18),
     *START(4),DDATE,DAYS,S,PN,LN,PLYS
      BYTE THK(8),GRADE(8),TYPE(5),EXT(5),WIDTH(2),PERNAM(4,6),
     *TIM(8),DAT(9),HEAD(4,33),DASH(132),DNAM(5),SNAM(7),
     *OPER(4,8),EQUAL(132),SHF(3,4)
      REAL DIFF,PRODQT,PR,RATE,FRC(7)
      COMMON /MODEL1/ MAXTK,CPTY,CONST,LEN,START,DAYS,DDATE,S,SHFT
      COMMON /MODEL2/ PRODQT,PRODID,SHIFT,MACHID,N,PNUM,DIFF,
     *PMACH,J,DAY,PERIOD,LFLAG,PQTY,SHF
      COMMON /MODEL3/ THK,GRADE,TYPE,EXT,WIDTH
      COMMON /MODEL4/ PERNAM,HEAD,DASH,DNAM,SNAM,OPER,EQUAL
C
C  CHECKING OPERATION HEADER.
C
      IF (J.GT.1) GO TO 60
C
C  SPREADER OPERATION
C
      READ (4'PRODID) PN,LN,PLYS,PR,CONST,LEN,(TYPE(M),M=1,3),
     Z(GRADE(M),M=1,8),(EXT(M),M=1,3),(THK(M),M=1,5)
      RETURN
C
C  CHECKING OPERATION COUNTER.
C
 60   IF (J.GT.3) GO TO 75
C
C  PATCHER/JOINTER OPERATION
C
      READ(4'PRODID) (THK(M),M=1,4),LEN,(GRADE(M),M=1,4),
     *(WIDTH(M),M=1,2)
      RETURN
C
C  DRYER OPERATION
C
 75   READ(4'PRODID) (FRC(M),M=1,8),RATE,(GRADE(M),M=1,4),
     *(THK(M),M=1,4),LEN,(WIDTH(M),M=1,2)
      RETURN
      END
```

```
C
C  DATE:  25 AUG. 1983
C
C  THIS SUBPROGRAM WRITES BLANK CHARACTER STRINGS IN ONE OF THE
C  OUTPUT FILES AS DESIGNATED BY THE FORMATTING VARIABLES.
C
C  AUTHOR:  ERIC SCHILD
C
      SUBROUTINE BLANKS
      INTEGER MAXTK(3,50),CONST,LEN,CPTY(4),LFLAG,PRODID,SHIFT,MACHID,
     *DAY,J,N,PNUM(18),SHFT(4),PQTY,P,MACH(18),PERIOD,PMACH(18),
     *START(4),DDATE,DAYS,S
      BYTE THK(8),GRADE(8),TYPE(5),EXT(5),WIDTH(2),PERNAM(4,6),
     *TIM(8),DAT(9),HEAD(4,33),DASH(132),DNAM(5),SNAM(7),
     *OPER(4,8),EQUAL(132),SHF(3,4),BLKS(33)
      REAL DIFF,PRODQT
      COMMON /MODEL1/ MAXTK,CPTY,CONST,LEN,START,DAYS,DDATE,S,SHFT
      COMMON /MODEL2/ PRODQT,PRODID,SHIFT,MACHID,N,PNUM,DIFF,
     *PMACH,J,DAY,PERIOD,LFLAG,PQTY,SHF
      COMMON /MODEL3/ THK,GRADE,TYPE,EXT,WIDTH
      COMMON /MODEL4/ PERNAM,HEAD,DASH,DNAM,SNAM,OPER,EQUAL
C
C  SETTING THE BOUND FOR THE LOOP USING THE FORMATTING VARIABLE
C
      L=PNUM(N)
      M=PNUM(N)+(MAXTK(SHIFT,DAY)-PMACH(N))-1
      DO 25 K=L,M
      I=4
 10   IF (N.GT.I) GO TO 15
      P=10+N-I
      ENCODE(33,40,BLKS(1))
      WRITE(P,35) (BLKS(JN),JN=1,33)
      GO TO 25
 15   I=I+4
      GO TO 10
 25   CONTINUE
C
C  RESET TASK COUNTER BY M PLUS ONE.
C
      PNUM(N)=M+1
 35   FORMAT (T2,33A1)
 40   FORMAT ('                                   ')
      RETURN
      END
```

```
C
C  DATE:  25 AUG. 1983
C
C  THIS SUBPROGRAM ADVANCES THE APPROPRIATE TIME (SHIFT AND DAY)
C  COUNTER AND WRITES THE SHIFT CHANGE HEADER IN ONE OF THE
C  FOUR OUTPUT FILES OPENED IN SUB OPENFL.
C
C  AUTHOR:  ERIC SCHILD
C
      SUBROUTINE HEADR
      INTEGER MAXTK(3,50),CONST,LEN,CPTY(4),LFLAG,PRODID,SHIFT,MACHID,
     *DAY,J,N,PNUM(18),SHFT(4),PQTY,P,MACH(18),PERIOD,PMACH(18),
     *START(4),DDATE,DAYS,S
      BYTE THK(9),GRADE(8),TYPE(5),EXT(5),WIDTH(2),PERNAM(4,6),
     *TIM(8),DAT(9),HEAD(4,33),DASH(132),DNAM(5),SNAM(7),
     *OPER(4,8),EQUAL(132),SHF(3,4),BLKS(33)
      REAL DIFF,PRODQT
      COMMON /MODEL1/ MAXTK,CPTY,CONST,LEN,START,DAYS,DDATE,S,SHFT
      COMMON /MODEL2/ PRODQT,PRODID,SHIFT,MACHID,N,PNUM,DIFF,
     *PMACH,J,DAY,PERIOD,LFLAG,PQTY,SHF
      COMMON /MODEL3/ THK,GRADE,TYPE,EXT,WIDTH
      COMMON /MODEL4/ PERNAM,HEAD,DASH,DNAM,SNAM,OPER,EQUAL
C
C  CHECK TO SEE IF A MACHINE CHANGE HAS OCCURRED. IF
C  SO, RESET SHIFT AND DAY TO ONE.
C
      IF (MACHID.NE.N) GO TO 15
C
C  INCREMENTING TIME (SHIFT AND DAY) COUNTER.
C
      IF (SHIFT.GE.3) GO TO 10
      SHIFT=SHIFT+1
      GO TO 55
 10   SHIFT=1
      DAY=DAY+1
      GO TO 55
C
C
 15   DAY=START(J)
      SHIFT=SHFT(J)
 55   PMACH(N)=1
C
      I=4
      ENCODE(33,65,BLKS(1))
C
C  CHECKING HEADER FLAG TO SEE IF HEADER IS REQUIRED
C
      IF (LFLAG.EQ.1) GO TO 60
C
C  CHECKING TO SEE WHICH OUTPUT FILE THE SHIFT HEADER WILL BE
C  PLACED-IF THE CURRENT MACHINE IS:
C
C              1,5,9,13,OR 17- SET L=1
C           2,6,10,14,OR 18-SET L=2
C              3,7,11,15,OR 19-SET L=3
C              4,8,12,16,OR 20-SET L=4
C
 5    IF (N.GT.I) GO TO 45
      L=10+N-I
      IF (L.GT.7) GO TO 30
```

```
      WRITE (L.25) (EQUAL(M),M=1.33)
      WRITE (L.25) (BLKS(M).M=1,33)
      WRITE (L,20) (DNAM(M),M=1,4).DAY,(SNAM(M),M=1.6),
     Z(PERNAM(SHIFT.M).M=1.6)
      WRITE (L,25) (DASH(M).M=1,33)
      GO TO 50
 30   WRITE (L.25) (EQUAL(M),M=1.33)
      WRITE(L.25) (BLKS(M).M=1,33)
      WRITE(L.25) (BLKS(M).M=1.33)
      WRITE (L,25) (DASH(M),M=1,33)
      GO TO 50
 45   I=I+4
      GO TO 5
 50   CONTINUE
C
C  INCREMENT TASK COUNTER BY FOUR TO COMPENSATE FOR THE FOUR LINE SHIFT
C  HEADER.
C
      PNUM(N)=PNUM(N)+4
      RETURN
 60   LFLAG=0
 20   FORMAT (T2.4A1.I3.1X,6A1.1X.6A1)
 25   FORMAT (T2.33A1)
 65   FORMAT ('                                        ')
      RETURN
      END
```

```
C
C   DATE:  25 AUG. 1983
C
C   THIS SUBPROGRAM WRITES THE TRANSLATED VERSION OF
C   EACH TASK IN THE APPROPRIATE OUTPUT FILE.
C
C   AUTHOR:  ERIC SCHILD
C
      SUBROUTINE CHCODE
      INTEGER MAXTK(3.50).CONST.LEN.CPTY(4).LFLAG,PRODID.SHIFT.MACHID.
     *DAY,J,N,PNUM(18).SHFT(4).PQTY.P.MACH(18).PERIOD.PMACH(18),
     *START(4).DDATE.DAYS.S
      BYTE THK(8).GRADE(8).TYPE(5).EXT(5).WIDTH(2).PERNAM(4.6).
     *TIM(8).DAT(9).HEAD(4,33),DASH(132),DNAM(5).SNAM(7),
     *OPER(4.8),EQUAL(132).SHF(3,4)
      REAL DIFF,PRODQT
      COMMON /MODEL1/ MAXTK.CPTY.CONST.LEN.START.DAYS.DDATE.S.SHFT
      COMMON /MODEL2/ PRODQT.PRODID.SHIFT.MACHID.N.PNUM.DIFF.
     *PMACH.J.DAY.PERIOD.LFLAG.PQTY.SHF
      COMMON /MODEL3/ THK,GRADE.TYPE.EXT.WIDTH
      COMMON /MODEL4/ PERNAM.HEAD.DASH.DNAM.SNAM.OPER.EQUAL
C
C   CHECKING TO DETERMINE WHICH FILE THE CURRENT TASK SHOULD BE
C   PLACED- IF THE CURRENT MACHINE IS:
C
C                             1.5.9.13.OR 17-SET L=1
C                             2,6,10,14.OR 18-SET L=2
C                             3.7,11,15,OR 19-SET L=3
C                             4.8,12,16.OR 20-SET L=4
C
      I=4
 10   IF (MACHID.GT.I) GO TO 35
      L=10+MACHID-I
C
C   CHECKING OPERATION COUNTER
C
      IF(J.GT.1) GO TO 30
C
C   SPREADER OPERATION
C
      WRITE (L.65) PQTY.
     *(THK(M).M=1,5).LEN.(TYPE(M).M=1.3).(GRADE(M).M=1.8).
     *(EXT(M).M=1.3).CONST
      RETURN
C
C   OTHER OPERATIONS
C
 30   WRITE (L.80) PRODQT.
     *(THK(M).M=1.4).LEN.(WIDTH(M).M=1.2).(GRADE(M).M=1.4).
     *DAYS.(SHF(S,M).M=1.4)
      RETURN
 35   I=I+4
      GO TO 10
 65   FORMAT(T2,I4,1X,5A1.I3.1X,3A1,1X.8A1.1X.3A1.I2.1X)
 80   FORMAT(T2,F6.2,1X.4A1.1X.I2.2X.2A1.2X.4A1.I3.1X,4A1.2X)
      RETURN
      END
```

```
C
C  DATE:  25 AUG. 1983
C
C  THIS SUBPROGRAM WRITES THE SHIFT AND MACHINE HEADERS, AND THE
C  ENTIRE CONTENT OF THE FOUR OUTPUT FILES CREATED IN THE EARLIER
C  ROUTINES
C
C  AUTHOR:  ERIC SCHILD
C
      SUBROUTINE RESLT
      INTEGER MAXTK(3,50),CONST,LEN,CPTY(4),LFLAG,PRODID,SHIFT,MACHID,
     *DAY,J,N,PNUM(18),SHFT(4),PQTY,P,MACH(18),PERIOD,PMACH(18),
     *START(4),DDATE,DAYS,S,MAX,NOM(5)
      BYTE THK(8),GRADE(8),TYPE(5),EXT(5),WIDTH(2),PERNAM(4,6),
     *TIM(8),DAT(9),HEAD(4,33),DASH(132),DNAM(5),SNAM(7),
     *OPER(4,8),EQUAL(132),SHF(3,4),ARRY(32,4)
      REAL DIFF,PRODQT
      COMMON /MODEL1/ MAXTK,CPTY,CONST,LEN,START,DAYS,DDATE,S,SHFT
      COMMON /MODEL2/ PRODQT,PRODID,SHIFT,MACHID,N,PNUM,DIFF,
     *PMACH,J,DAY,PERIOD,LFLAG,PQTY,SHF
      COMMON /MODEL3/ THK,GRADE,TYPE,EXT,WIDTH
      COMMON /MODEL4/ PERNAM,HEAD,DASH,DNAM,SNAM,OPER,EQUAL
C
      CALL ASSIGN(7,'SY:MACH15.DAT')
      CALL ASSIGN(8,'SY:MACH26.DAT')
      CALL ASSIGN(9,'SY:MACH37.DAT')
      CALL ASSIGN(10,'SY:MACH48.DAT')
C
C  ESTABLISHING THE MAXIMUM NUMBER OF ENTRIES AMONG ALL THE MACHINES
C  SO THAT THE ENTIRE CONTENT OF EACH  OUTPUT FILE WILL
C  BE PRINTED.
C
      MAX=PNUM(1)-1
C
C  DETERMINING THE NUMBER OF SETS OF FOUR MACHINES THAT WILL BE
C  REQUIRED FOR THE CURRENT OPERATION
C
      IF (CPTY(J).GT.4) GO TO 5
      M=1
      NOM(1)=CPTY(J)
      GO TO 25
 5    IF (CPTY(J).GT.8) GO TO 10
      M=2
      NOM(1)=4
      NOM(2)=CPTY(J)-4
      GO TO 25
 10   IF(CPTY(J).GT.12)GO TO 15
      M=3
      NOM(1)=4
      NOM(2)=4
      NOM(3)=CPTY(J)-8
      GO TO 25
15    IF(CPTY(J).GT.16)GO TO 20
      M=4
      NOM(1)=4
      NOM(2)=4
      NOM(3)=4
      NOM(4)=CPTY(J)-12
      GO TO 25
20    M=5
```

```
      NOM(1)=4
      NOM(2)=4
      NOM(3)=4
      NOM(4)=4
      NOM(5)=CPTY(J)-16
C
C
 25   DO 75 P=1,M
C
C  SHIFT HEADER.
C
      L=1+4*(P-1)
      WRITE (6,30) (N,N=L,L+NOM(P)-1)
 30   FORMAT (T2,'MACHINE:'/T2,I16,3I33)
      WRITE (6,35) START(J),(PERNAM(SHFT(J),N),N=1,6),
 35   FORMAT ('0',T2,'DAY: ',I2,' SHIFT: ',6A1/T2,131A1)
C
C  MACHINE HEADER.
C
      WRITE (6,40) ((HEAD(J,N),N=1,32),I=1,NOM(P))
 40   FORMAT (T2,32A1,'|',32A1,'|',32A1,'|',32A1)
C
C
 45   DO 70 K=1,MAX
      IF (NOM(P).GT.1) GO TO 50
      READ(7,85,END=70) (ARRY(N,1),N=1,32)
      L=1
      GO TO 65
C
 50   IF (NOM(P).GT.2) GO TO 55
      READ(7,85,END=70) (ARRY(N,1),N=1,32)
      READ(8,85,END=65) (ARRY(N,2),N=1,32)
      L=2
      GO TO 65
C
 55   IF (NOM(P).GT.3) GO TO 60
      READ (7,85,END=70) (ARRY(N,1),N=1,32)
      READ (8,85,END=65) (ARRY(N,2),N=1,32)
      READ (9,85,END=65) (ARRY(N,3),N=1,32)
      L=3
      GO TO 65
C
 60   READ(7,85,END=70) (ARRY(N,1),N=1,32)
      READ(8,85,END=65) (ARRY(N,2),N=1,32)
      READ(9,85,END=65) (ARRY(N,3),N=1,32)
      READ(10,85,END=65) (ARRY(N,4),N=1,32)
      L=4
C
C  WRITING THE CONTENTS OF THE FILES.
C
 65   WRITE (6,90) ((ARRY(N,KN),N=1,32),KN=1,L)
C
 70   CONTINUE
      WRITE (6,80) (EQUAL(N),N=1,131)
 75   CONTINUE
      CALL CLOSE(7)
      CALL CLOSE(8)
      CALL CLOSE(9)
      CALL CLOSE(10)
 80   FORMAT (T2,131A1)
```

```
85    FORMAT (T2,33A1)
90    FORMAT (T2,32A1,'|',32A1,'|',32A1,'|',32A1)
      RETURN
      END
```

APPENDIX B

DATA FILES

APPENDIX B.1 DESCRIPTION OF DATA FILES

B.1.1 Program Generated Files

DRYTSK.DAT- equivalent to PATTSK.DAT except this file represents the sequenced list of tasks assigned to the dry end operation; created in phase two; sequential access.

ENTRY1.DAT- this file develops the formatting statistics for each shift of the spreader operation; used in phase three to assist subroutine BLANKS in inserting the correct number of blanks between the last task written for the previous shift, and the new shift header; created in phase one; sequential access.

ENTRY2.DAT- identical to ENTRY1.DAT except the formatting statistics are for each shift of the patcher operation; created in phase two; sequential access.

ENTRY3.DAT- identical to ENTRY1.DAT except the formatting statistics are for each shift of the jointer operation; created in phase two; sequential access.

ENTRY4.DAT- identical to ENTRY1.DAT except the formatting statistics are for each shift of the dryer operation; created in phase two; sequential access.

JOITSK.DAT- equivalent to PATTSK.DAT except this file represents the sequenced list of tasks assigned to the jointer operation; created in phase two; sequential access.

MACH15.DAT- file created and used in phase three to store the formatted output for tasks assigned to machines one and five of any operation; sequential access.

MACH26.DAT - file created and used in phase three to store the formatted output for tasks assigned to machine two and six of any operation; sequential access.

MACH37.DAT - file created and used in phase three to store the formatted output for tasks assigned to machine three and seven of any operation; sequential access.

MACH48.DAT - file created and used in phase three to store the formatted output for tasks assigned to machine four and eight of any operation; sequential access.

PAT.DAT - master production schedule; created in phase one;

sequential access.

PATTSK.DAT- sequenced list of tasks that require processing on the patcher operation; created in phase two; sequential access.

SEQ1.DAT - used to pass the list of sequenced orders for the spreaders to phase three, where they will be formatted (using ENTRY1.DAT) as illustrated in appendix C; created in phase one; sequential access.

SEQ2.DAT - equivalent to SEQ1.DAT except this file represents the list of sequenced tasks for the patcher operation. The associated formatting file is ENTRY2.DAT; created in phase ; sequential access.

SEQ3.DAT - equivalent to SEQ1.DAT except this file represents the list of sequenced tasks for the jointer operation. The associated formatting file is ENTRY3.DAT; created in phase two; sequential access.

SEQ4.DAT - equivalent to SEQ1.DAT except this file represents the list of sequenced tasks for the dryer operation. The associated formatting file is ENTRY4.DAT; created in phase two; sequential access.

START.DAT - file of the start times for the four operations; created in phase two and used in phase three; sequential access.

B.1.2 Archival files

DRYCOD.DAT - consists of the numeric identification codes that defines each entry of DRYDEV.DAT; assists in accessing entries from DRYDEV.DAT.

DRYDEV.DAT - gives the processing rates and the grade breakdown that result when drying a green veneer grade; direct access.

DVEN.DAT - this file is identical to INVDEV.DAT but it is used in phase one to translate the initial dry veneer inventory file, DRY.DAT; sequential access.

GRADE.DAT - used in conjunction with both DRYDEV.DAT and PATDEV.DAT to identify the grades that result when processing the specified veneer grade; sequential access.

INVCOD.DAT- used in phase two to set up the linked inventory array; contains the numeric identification codes for each entry of INVDEV.DAT; sequential access.

INVDEV.DAT- used in phase three to translate the numeric identification code to a more meaningful character strings for the veneer tasks in SEQ2.DAT and SEQ4.DAT; direct access.

JRATE.DAT - consists of the processing rates for the jointer operation; sequential access.

PANEL.DAT - consists of the current panel product line. Each panel is defined with a character description, a processing rate on the spreaders , and the bill of material that will be used in the construction of the panel; direct access.

PANUM.DAT - used to identify each entry of PANEL.DAT using a six digit identification code; sequential access.

PATCOD.DAT - consists of the numeric identification codes that define each entry of PATDEV.DAT; sequential access.

PATDEV.DAT - gives the processing rates and the grade breakdowns that result when patching a patchable veneer grade; direct access.

PER.DAT - contains the column headers that will be used when writing the report in phase three; sequential access.

B.1.3 User-defined Policy Files

DPOLI.DAT - equivalent to JPOLI.DAT except used in allocating the tasks among the seven machines of the dryer operation; sequential access.

GLPOLI.DAT - use to designate the machine restriction s that must be considered when assigning the orders among the six machines of the spreader operation; direct access.

JPOLI.DAT - used to clarify the machine restrictions and any management constraints that may be necessary when allocating tasks among the five machines of the jointer operation; sequential access.

MIDLE.DAT - used to designate which shifts will be idle for the machines of the spreader operation.

STIME.DAT -specifies those panel grades that should definitely not be sequenced in consecutive order; sequential access.

VENDEV.DAT- veneer development file; incorporates the users policies for developing veneer grades from other veneer grade sources; direct access.

B.1.4 User-defined Input Files

DRY.DAT - used to set up the initial dry inventory for a given run of the scheduling package; each inventory item is input with a numeric code and quantity; sequential access.

TASK.DAT - contains the panel orders to be scheduled; each order is defined with a numeric identification code and the order quantity; sequential access.

VENDEV. DAT Veneer Development File

Column headings (as printed): veneer code; no. of full width alternative sources; Alt. 1: veneer grade, patchable, no. of sources, source 1, source 2; Alt. 2: veneer grade, patchable, no. of sources, source 1, source 2; Alt. 3: veneer grade, patchable, no. of sources, source1, source2; Alt. 1: no. of widths, width 1, width 2, width 3; Alt. 2: no. of widths, width 1, width 3, width 3; Alt. 3: no. of widths, width 1, width 2, width 3; no. of narrow widths; alternative 1; alternative 2; alternative 3; no. of widths, width 1, width 2, width 3; no. of widths, width 1, width 2, width 3; no. of widths, width 1, width 2, width 3

111	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
112	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
113	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
114	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
121	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
122	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
123	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
124	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
131	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
132	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
133	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
134	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
141	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
142	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
143	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
144	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
211	2	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	1	2	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
212	2	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	1	2	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
213	2	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	1	2	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
214	2	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	1	2	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
221	2	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	1	2	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
222	2	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	1	2	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
223	2	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	1	2	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
224	2	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	1	2	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
231	2	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	1	2	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
232	2	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	1	2	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
233	2	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	1	2	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
234	2	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	1	2	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
241	2	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	1	2	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
242	2	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	1	2	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
243	2	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	1	2	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
244	2	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	1	2	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
311	1	9	1	2	3	4	0	0	0	0	0	0	0	0	0	0	3	1	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
312	1	9	1	2	3	4	0	0	0	0	0	0	0	0	0	0	3	1	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
313	1	9	1	2	3	4	0	0	0	0	0	0	0	0	0	0	3	1	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
314	1	9	1	2	3	4	0	0	0	0	0	0	0	0	0	0	3	1	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
321	1	9	1	2	3	4	0	0	0	0	0	0	0	0	0	0	3	1	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
322	1	9	1	2	3	4	0	0	0	0	0	0	0	0	0	0	3	1	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
323	1	9	1	2	3	4	0	0	0	0	0	0	0	0	0	0	3	1	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
324	1	9	1	2	3	4	0	0	0	0	0	0	0	0	0	0	3	1	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
331	1	9	1	2	3	4	0	0	0	0	0	0	0	0	0	0	3	1	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
332	1	9	1	2	3	4	0	0	0	0	0	0	0	0	0	0	3	1	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
333	1	9	1	2	3	4	0	0	0	0	0	0	0	0	0	0	3	1	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
334	1	9	1	2	3	4	0	0	0	0	0	0	0	0	0	0	3	1	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
341	1	9	1	2	3	4	0	0	0	0	0	0	0	0	0	0	3	1	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
342	1	9	1	2	3	4	0	0	0	0	0	0	0	0	0	0	3	1	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
343	1	9	1	2	3	4	0	0	0	0	0	0	0	0	0	0	3	1	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
344	1	9	1	2	3	4	0	0	0	0	0	0	0	0	0	0	3	1	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
411	1	9	1	2	3	4	0	0	0	0	0	0	0	0	0	0	3	1	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
412	1	9	1	2	3	4	0	0	0	0	0	0	0	0	0	0	3	1	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
413	1	9	1	2	3	4	0	0	0	0	0	0	0	0	0	0	3	1	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
414	1	9	1	2	3	4	0	0	0	0	0	0	0	0	0	0	3	1	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
421	1	9	1	2	3	4	0	0	0	0	0	0	0	0	0	0	3	1	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
422	1	9	1	2	3	4	0	0	0	0	0	0	0	0	0	0	3	1	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
423	1	9	1	2	3	4	0	0	0	0	0	0	0	0	0	0	3	1	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
424	1	9	1	2	3	4	0	0	0	0	0	0	0	0	0	0	3	1	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
431	1	9	1	2	3	4	0	0	0	0	0	0	0	0	0	0	3	1	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
432	1	9	1	2	3	4	0	0	0	0	0	0	0	0	0	0	3	1	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
433	1	9	1	2	3	4	0	0	0	0	0	0	0	0	0	0	3	1	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
434	1	9	1	2	3	4	0	0	0	0	0	0	0	0	0	0	3	1	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
441	1	9	1	2	3	4	0	0	0	0	0	0	0	0	0	0	3	1	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
442	1	9	1	2	3	4	0	0	0	0	0	0	0	0	0	0	3	1	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
443	1	9	1	2	3	4	0	0	0	0	0	0	0	0	0	0	3	1	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
444	1	9	1	2	3	4	0	0	0	0	0	0	0	0	0	0	3	1	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VENDEV. DAT cont'd

```
511   3 5 0 0 0 0 6 1 1 5 0 7 0 0 0 0 2 2 3 0 3 1 2 3 2 2 3 0 3 51011 2 2 3 0 3 1 2 3 1 3 0 0
512   3 5 0 0 0 0 6 1 1 5 0 7 0 0 0 0 2 2 3 0 3 1 2 3 2 2 3 0 3 51011 2 2 3 0 3 1 2 3 1 3 0 0
513   3 5 0 0 0 0 6 1 1 5 0 7 0 0 0 0 2 2 3 0 3 1 2 3 2 2 3 0 3 51011 2 2 3 0 3 1 2 3 1 3 0 0
514   3 5 0 0 0 0 6 1 1 5 0 7 0 0 0 0 2 2 3 0 3 1 2 3 2 2 3 0 3 51011 2 2 3 0 3 1 2 3 1 3 0 0
521   3 5 0 0 0 0 6 1 1 5 0 7 0 0 0 0 2 2 3 0 3 1 2 3 2 2 3 0 3 51011 2 2 3 0 3 1 2 3 1 3 0 0
522   3 5 0 0 0 0 6 1 1 5 0 7 0 0 0 0 2 2 3 0 3 1 2 3 2 2 3 0 3 51011 2 2 3 0 3 1 2 3 1 3 0 0
523   3 5 0 0 0 0 6 1 1 5 0 7 0 0 0 0 2 2 3 0 3 1 2 3 2 2 3 0 3 51011 2 2 3 0 3 1 2 3 1 3 0 0
524   3 5 0 0 0 0 6 1 1 5 0 7 0 0 0 0 2 2 3 0 3 1 2 3 2 2 3 0 3 51011 2 2 3 0 3 1 2 3 1 3 0 0
531   3 5 0 0 0 0 6 1 1 5 0 7 0 0 0 0 2 2 3 0 3 1 2 3 2 2 3 0 3 51011 2 2 3 0 3 1 2 3 1 3 0 0
532   3 5 0 0 0 0 6 1 1 5 0 7 0 0 0 0 2 2 3 0 3 1 2 3 2 2 3 0 3 51011 2 2 3 0 3 1 2 3 1 3 0 0
533   3 5 0 0 0 0 6 1 1 5 0 7 0 0 0 0 2 2 3 0 3 1 2 3 2 2 3 0 3 51011 2 2 3 0 3 1 2 3 1 3 0 0
534   3 5 0 0 0 0 6 1 1 5 0 7 0 0 0 0 2 2 3 0 3 1 2 3 2 2 3 0 3 51011 2 2 3 0 3 1 2 3 1 3 0 0
541   3 5 0 0 0 0 6 1 1 5 0 7 0 0 0 0 2 2 3 0 3 1 2 3 2 2 3 0 3 51011 2 2 3 0 3 1 2 3 1 3 0 0
542   3 5 0 0 0 0 6 1 1 5 0 7 0 0 0 0 2 2 3 0 3 1 2 3 2 2 3 0 3 51011 2 2 3 0 3 1 2 3 1 3 0 0
543   3 5 0 0 0 0 6 1 1 5 0 7 0 0 0 0 2 2 3 0 3 1 2 3 2 2 3 0 3 51011 2 2 3 0 3 1 2 3 1 3 0 0
544   3 5 0 0 0 0 6 1 1 5 0 7 0 0 0 0 2 2 3 0 3 1 2 3 2 2 3 0 3 51011 2 2 3 0 3 1 2 3 1 3 0 0
611   1 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 3 0 0 0 0 0 0 0 0 0 3 61011 2 2 3 0 3 1 2 3 1 3 0 0
612   1 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 3 0 0 0 0 0 0 0 0 0 3 61011 2 2 3 0 3 1 2 3 1 3 0 0
613   1 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 3 0 0 0 0 0 0 0 0 0 3 61011 2 2 3 0 3 1 2 3 1 3 0 0
614   1 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 3 0 0 0 0 0 0 0 0 0 3 61011 2 2 3 0 3 1 2 3 1 3 0 0
621   1 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 3 0 0 0 0 0 0 0 0 0 3 61011 2 2 3 0 3 1 2 3 1 3 0 0
622   1 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 3 0 0 0 0 0 0 0 0 0 3 61011 2 2 3 0 3 1 2 3 1 3 0 0
623   1 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 3 0 0 0 0 0 0 0 0 0 3 61011 2 2 3 0 3 1 2 3 1 3 0 0
624   1 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 3 0 0 0 0 0 0 0 0 0 3 61011 2 2 3 0 3 1 2 3 1 3 0 0
631   1 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 3 0 0 0 0 0 0 0 0 0 3 61011 2 2 3 0 3 1 2 3 1 3 0 0
632   1 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 3 0 0 0 0 0 0 0 0 0 3 61011 2 2 3 0 3 1 2 3 1 3 0 0
633   1 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 3 0 0 0 0 0 0 0 0 0 3 61011 2 2 3 0 3 1 2 3 1 3 0 0
634   1 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 3 0 0 0 0 0 0 0 0 0 3 61011 2 2 3 0 3 1 2 3 1 3 0 0
641   1 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 3 0 0 0 0 0 0 0 0 0 3 61011 2 2 3 0 3 1 2 3 1 3 0 0
642   1 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 3 0 0 0 0 0 0 0 0 0 3 61011 2 2 3 0 3 1 2 3 1 3 0 0
643   1 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 3 0 0 0 0 0 0 0 0 0 3 61011 2 2 3 0 3 1 2 3 1 3 0 0
644   1 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 3 0 0 0 0 0 0 0 0 0 3 61011 2 2 3 0 3 1 2 3 1 3 0 0
711   1 9 1 2 7 4 0 0 0 0 0 0 0 0 0 0 3 1 2 3 0 0 0 0 0 0 0 0 111 0 0 1 3 0 0 0 0 0 0 0 0 0 0
712   1 9 1 2 7 4 0 0 0 0 0 0 0 0 0 0 3 1 2 3 0 0 0 0 0 0 0 0 111 0 0 1 3 0 0 0 0 0 0 0 0 0 0
713   1 9 1 2 7 4 0 0 0 0 0 0 0 0 0 0 3 1 2 3 0 0 0 0 0 0 0 0 111 0 0 1 3 0 0 0 0 0 0 0 0 0 0
714   1 9 1 2 7 4 0 0 0 0 0 0 0 0 0 0 3 1 2 3 0 0 0 0 0 0 0 0 111 0 0 1 3 0 0 0 0 0 0 0 0 0 0
721   1 7 1 1 7 0 0 0 0 0 0 0 0 0 0 0 1 2 0 0 0 0 0 0 0 0 0 0 111 0 0 1 3 0 0 0 0 0 0 0 0 0 0
722   1 7 1 1 7 0 0 0 0 0 0 0 0 0 0 0 1 2 0 0 0 0 0 0 0 0 0 0 111 0 0 1 3 0 0 0 0 0 0 0 0 0 0
723   1 7 1 1 7 0 0 0 0 0 0 0 0 0 0 0 1 2 0 0 0 0 0 0 0 0 0 0 111 0 0 1 3 0 0 0 0 0 0 0 0 0 0
724   1 7 1 1 7 0 0 0 0 0 0 0 0 0 0 0 1 2 0 0 0 0 0 0 0 0 0 0 111 0 0 1 3 0 0 0 0 0 0 0 0 0 0
731   1 9 1 2 7 4 0 0 0 0 0 0 0 0 0 0 3 1 2 3 0 0 0 0 0 0 0 0 111 0 0 1 3 0 0 0 0 0 0 0 0 0 0
732   1 9 1 2 7 4 0 0 0 0 0 0 0 0 0 0 3 1 2 3 0 0 0 0 0 0 0 0 111 0 0 1 3 0 0 0 0 0 0 0 0 0 0
733   1 9 1 2 7 4 0 0 0 0 0 0 0 0 0 0 3 1 2 3 0 0 0 0 0 0 0 0 111 0 0 1 3 0 0 0 0 0 0 0 0 0 0
734   1 9 1 2 7 4 0 0 0 0 0 0 0 0 0 0 3 1 2 3 0 0 0 0 0 0 0 0 111 0 0 1 3 0 0 0 0 0 0 0 0 0 0
741   1 9 1 2 7 4 0 0 0 0 0 0 0 0 0 0 3 1 2 3 0 0 0 0 0 0 0 0 111 0 0 1 3 0 0 0 0 0 0 0 0 0 0
742   1 9 1 2 7 4 0 0 0 0 0 0 0 0 0 0 3 1 2 3 0 0 0 0 0 0 0 0 111 0 0 1 3 0 0 0 0 0 0 0 0 0 0
743   1 9 1 2 7 4 0 0 0 0 0 0 0 0 0 0 3 1 2 3 0 0 0 0 0 0 0 0 111 0 0 1 3 0 0 0 0 0 0 0 0 0 0
744   1 9 1 2 7 4 0 0 0 0 0 0 0 0 0 0 3 1 2 3 0 0 0 0 0 0 0 0 111 0 0 1 3 0 0 0 0 0 0 0 0 0 0
811   2 1 1 1 8 0 2 1 1 8 0 0 0 0 0 0 1 1 0 0 2 1 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
812   2 1 1 1 8 0 2 1 1 8 0 0 0 0 0 0 1 1 0 0 2 1 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
813   2 1 1 1 8 0 2 1 1 8 0 0 0 0 0 0 1 1 0 0 2 1 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
814   2 1 1 1 8 0 2 1 1 8 0 0 0 0 0 0 1 1 0 0 2 1 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
821   2 1 1 1 8 0 2 1 1 8 0 0 0 0 0 0 1 1 0 0 2 1 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
822   2 1 1 1 8 0 2 1 1 8 0 0 0 0 0 0 1 1 0 0 2 1 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
823   2 1 1 1 8 0 2 1 1 8 0 0 0 0 0 0 1 1 0 0 2 1 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
824   2 1 1 1 8 0 2 1 1 8 0 0 0 0 0 0 1 1 0 0 2 1 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
831   2 1 1 1 8 0 2 1 1 8 0 0 0 0 0 0 1 1 0 0 2 1 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
832   2 1 1 1 8 0 2 1 1 8 0 0 0 0 0 0 1 1 0 0 2 1 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
833   2 1 1 1 8 0 2 1 1 8 0 0 0 0 0 0 1 1 0 0 2 1 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
834   2 1 1 1 8 0 2 1 1 8 0 0 0 0 0 0 1 1 0 0 2 1 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
841   2 1 1 1 8 0 2 1 1 8 0 0 0 0 0 0 1 1 0 0 2 1 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
842   2 1 1 1 8 0 2 1 1 8 0 0 0 0 0 0 1 1 0 0 2 1 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
843   2 1 1 1 8 0 2 1 1 8 0 0 0 0 0 0 1 1 0 0 2 1 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
844   2 1 1 1 8 0 2 1 1 8 0 0 0 0 0 0 1 1 0 0 2 1 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

GLPOLI. DAT — Machine Restriction File

Machine	grade 1	grade 2	grade 3	grade 4	grade 5	length priority 1	length priority 2	length priority 3	length priority 4
1	1	3	2	5	5	1	1	1	1
2	1	3	2	5	5	2	3	4	1
3	2	3	1	5	5	3	2	4	4
4	4	3	2	1	1	1	1	1	1
5	2	3	1	5	5	3	2	4	4
6	4	3	2	1	5	4	3	2	1

(Column heads as printed: grade priority over columns 1–5; length priority over columns 1–4.)

JPOLI. DAT — Jointer Machine Restriction File

Columns: grade restriction / thickness restriction / length restriction / width restriction

Machine	
1	5234789000000000000000000000
2	5234789000000032344440000000
3	0000000000000000000000000000
4	0000000000000014444440000000
5	0000000000000023444440000000

DPOLI.DAT — Dryer Machine Restriction File

Columns: grade restriction / thickness restriction / length restriction / width restriction

Machine	
1	0000000133333323444440000000
2	0000000133333323444440000000
3	0000000000000024555550000000
4	0000000000000015555550000000
5	0000000000000015555550000000
6	0000000000000015555550000000
7	0000000000000014444440000000

JRATE. DAT — Jointer Production Rate File

thickness	length	rate
1	1	.03768
1	2	.03758
1	3	.03200
1	4	.02762
2	1	.03087
2	2	.02718
2	3	.02677
2	4	.02310
3	1	.01995
3	2	.01855
3	3	.01683
3	4	.01498
4	1	.01654
4	2	.01470
4	3	.01323
4	4	.01102

Bill of Material

panel identification code	no. of plys	production rate	construction	length	category	grade	type	thickness	Quantity: face	Quantity: back	Quantity: center	Quantity: center/crossband	Quantity: crossband	Grade: face	Grade: back	Grade: center	Grade: center/crossband	Grade: crossband
111121	3	4.17	1	8	SID	REGULAR	EXT	3/8	1	1	0	0	1	131	531	51	51	52
111131	5	2.29	1	8	SID	REGULAR	EXT	1/2	1	1	1	0	2	141	541	751	51	53
111132	5	2.29	2	8	SID	REGULAR	EXT	1/2	1	1	1	0	2	131	531	751	51	53
111141	5	2.08	1	8	SID	REGULAR	EXT	5/8	1	1	1	0	2	131	531	751	51	75
111142	5	2.08	2	8	SID	REGULAR	EXT	5/8	1	1	1	0	2	131	531	721	51	56
111221	3	4.17	1	9	SID	REGULAR	EXT	3/8	1	1	0	0	1	132	532	51	51	52
111231	5	2.08	1	9	SID	REGULAR	EXT	1/2	1	1	1	0	2	132	532	722	51	53
111241	5	2.08	1	9	SID	REGULAR	EXT	5/8	1	1	1	0	2	132	532	722	51	56
111321	3	3.50	1	10	SID	REGULAR	EXT	3/8	1	1	0	0	1	133	533	51	51	52
111331	5	1.75	1	10	SID	REGULAR	EXT	1/2	1	1	1	0	2	133	533	723	51	53
111341	5	1.75	1	10	SID	REGULAR	EXT	5/8	1	1	1	0	2	133	533	723	51	56
121141	5	2.08	1	8	SID	PREM-SCR	EXT	5/8	1	1	1	0	2	231	531	751	51	55
121142	5	2.08	2	8	SID	PREM-SCR	EXT	5/8	1	1	1	0	2	231	531	721	51	56
121143	5	2.08	3	8	SID	PREM-SCR	EXT	5/8	1	1	1	0	2	231	531	771	51	52
121241	5	2.08	1	9	SID	PREM-SCR	EXT	5/9	1	1	1	0	2	232	532	722	51	56
121242	5	2.08	2	9	SID	PREM-SCR	EXT	5/8	1	1	1	0	2	232	532	772	51	52
121341	5	1.75	1	10	SID	PREM-SCR	EXT	5/8	1	1	1	0	2	233	533	723	51	56
121441	5	1.33	1	12	SID	PREM-SCR	EXT	5/8	1	1	1	0	2	234	534	724	51	56
131141	5	2.08	1	8	SID	PREM-CCR	EXT	5/8	1	1	1	0	2	231	531	551	51	55
131142	5	2.08	2	8	SID	PREM-CCR	EXT	5/8	1	1	1	0	2	231	531	521	51	56
131143	5	2.08	3	8	SID	PREM-CCR	EXT	5/8	1	1	1	0	2	231	531	571	51	52
131241	5	2.08	1	9	SID	PREM-CCR	EXT	5/8	1	1	1	0	2	232	532	522	51	56
131242	5	2.08	2	9	SID	PREM-CCR	EXT	5/8	1	1	1	0	2	232	532	572	51	52
131341	5	1.75	1	10	SID	PREM-CCR	EXT	5/8	1	1	1	0	2	233	533	523	51	56
131441	5	1.33	1	12	SID	PREM-CCR	EXT	5/8	1	1	1	0	2	234	534	524	51	56
141141	5	2.08	1	8	SID	SND-SCTR	EXT	5/8	1	1	1	0	2	331	531	751	51	55
141142	5	2.08	2	8	SID	SND-SCTR	EXT	5/8	1	1	1	0	2	331	531	721	51	56
141143	5	2.08	3	8	SID	SND-SCTR	EXT	5/8	1	1	1	0	2	331	531	771	51	52
141241	5	2.08	1	9	SID	SND-SCTR	EXT	5/8	1	1	1	0	2	332	532	722	51	56
141242	5	1.85	2	9	SID	SND-SCTR	EXT	5/8	1	1	1	0	2	332	532	772	51	52
141341	5	1.75	1	10	SID	SND-SCTR	EXT	5/8	1	1	1	0	2	333	533	723	51	56
141441	5	1.33	1	12	SID	SND-SCTR	EXT	5/8	1	1	1	0	2	334	534	724	51	56
151141	5	2.08	1	8	SID	SND-CCTR	EXT	5/8	1	1	1	0	2	331	531	551	51	55
151142	5	2.08	2	8	SID	SND-CCTR	EXT	5/8	1	1	1	0	2	331	531	521	51	56
151143	5	2.08	3	8	SID	SND-CCTR	EXT	5/8	1	1	1	0	2	331	531	511	51	53
151241	5	2.08	1	9	SID	SND-CCTR	EXT	5/8	1	1	1	0	2	332	532	522	51	56
151242	5	2.08	2	9	SID	SND-CCTR	EXT	5/8	1	1	1	0	2	332	532	512	51	53
151243	5	2.08	3	9	SID	SND-CCTR	EXT	5/8	1	1	1	0	2	332	532	572	51	52
151341	5	1.75	1	10	SID	SND-CCTR	EXT	5/8	1	1	1	0	2	333	533	523	51	56
151342	5	1.75	2	10	SID	SND-CCTR	EXT	5/8	1	1	1	0	2	333	533	513	51	53
151441	5	1.33	1	12	SID	SND-CCTR	EXT	5/8	1	1	1	0	2	334	534	524	51	56
151442	5	1.33	2	12	SID	SND-CCTR	EXT	5/8	1	1	1	0	2	334	534	514	51	53
161121	3	4.17	1	8	SID	PREMIUM	EXT	3/8	1	1	0	0	1	231	531	51	51	52
161221	3	4.17	1	9	SID	PREMIUM	EXT	3/8	1	1	0	0	1	232	532	51	51	52
161321	3	3.50	1	10	SID	PREMIUM	EXT	3/8	1	1	0	0	1	233	533	51	51	52
161421	3	2.92	1	12	SID	PREMIUM	EXT	3/8	1	1	0	0	1	234	534	51	51	52
171121	3	4.17	1	8	SID	SND	EXT	3/8	1	1	0	0	1	331	531	51	51	52
171221	3	4.17	1	9	SID	SND	EXT	3/8	1	1	0	0	1	332	532	51	51	52
171321	3	3.50	1	10	SID	SND	EXT	3/8	1	1	0	0	1	333	533	51	51	52
171422	3	2.92	2	12	SID	SND	EXT	3/8	1	1	0	0	1	334	534	51	51	52
181131	5	2.29	1	8	SID	REGL-BNK	EXT	1/2	1	1	1	0	2	141	641	751	51	53
181132	5	2.29	2	8	SID	REGL-BNK	EXT	1/2	1	1	1	0	2	131	631	751	51	53
181133	5	2.29	3	8	SID	REGL-BNK	EXT	1/2	1	1	1	0	2	131	631	431	51	57
181231	5	2.04	1	9	SID	REGL-BNK	EXT	1/2	1	1	1	0	2	132	532	522	51	53
181331	5	1.83	1	10	SID	REGL-BNK	EXT	1/2	1	1	1	0	2	133	523	523	51	53
111101	3	4.17	1	8	SID	REGULAR	EXT	5/16	1	1	0	0	1	131	531	51	51	53
111201	3	4.17	1	9	SID	REGULAR	EXT	5/16	1	1	0	0	1	132	532	51	51	53
111301	3	3.50	1	10	SID	REGULAR	EXT	5/16	1	1	0	0	1	133	533	51	51	53
141131	5	2.08	1	8	SID	SND-SCTR	EXT	1/2	1	1	1	0	2	331	531	721	51	54
151132	5	2.08	2	8	SID	SND-CCTR	EXT	1/2	1	1	1	0	2	331	531	521	51	54
141331	5	1.75	1	10	SID	SND-SCTR	EXT	1/2	1	1	1	0	2	333	533	723	51	54
151331	5	1.75	1	10	SID	SND-CCTR	EXT	1/2	1	1	1	0	2	333	533	523	51	54
121231	5	2.08	1	9	SID	PREM-CCR	EXT	1/2	1	1	1	0	2	232	532	522	51	54
212131	5	2.08	1	8	SAN	AA	INT	1/2	1	1	1	0	2	331	331	621	51	63
212151	5	2.08	1	3	SAN	AA	INT	3/4	1	1	1	0	2	331	331	621	51	61
212331	5	1.75	1	10	SAN	AA	INT	1/2	1	1	1	0	2	333	333	623	51	63
212351	5	1.67	1	10	SAN	AA	INT	3/4	1	1	1	0	2	333	333	623	51	61
221111	3	4.17	1	8	SAN	AB	EXT	1/4	1	1	0	0	1	331	431	51	51	53
221121	3	4.17	1	8	SAN	AB	EXT	3/8	1	1	0	0	1	331	431	51	51	51
221131	5	2.08	1	8	SAN	AB	EXT	1/2	1	1	1	0	2	331	431	521	51	53
221132	5	2.08	2	8	SAN	AB	EXT	1/2	1	1	1	0	2	331	431	531	51	56
221141	5	2.08	1	8	SAN	AB	EXT	5/8	1	1	1	0	2	331	431	521	51	52
221151	5	2.08	1	8	SAN	AB	EXT	3/4	1	1	1	0	2	331	431	521	51	51
221161	7	1.46	1	8	SAN	AB	EXT	1 0/0	1	1	1	1	3	331	431	521	511	52
221311	3	3.50	1	10	SAN	AB	EXT	1/4	1	1	0	0	1	333	433	51	51	53
221321	3	3.50	1	10	SAN	AB	EXT	3/8	1	1	0	0	1	333	433	51	51	51

PANEL.DAT cont'd

221331	5	1.75	1	10	SAN	AB	EXT		1/2	1	1	1	0	2	333	433	523	51	53
221332	5	1.75	2	10	SAN	AB	EXT		1/2	1	1	1	0	2	333	433	533	51	56
221341	5	1.75	1	10	SAN	AB	EXT		5/8	1	1	1	0	2	333	433	523	51	52
221351	5	1.75	1	10	SAN	AB	EXT		3/4	1	1	1	0	2	333	433	523	51	51
221361	7	1.25	1	10	SAN	AB	EXT	1	0/0	1	1	1	1	3	333	433	523	513	52
221371	7	1.25	1	10	SAN	AB	EXT	1	1/8	1	1	2	0	3	333	433	623	61	51
221381	7	1.25	1	10	SAN	AB	EXT	1	1/4	1	1	2	0	3	323	423	523	51	61
221391	9	0.83	1	10	SAN	AB	EXT	1	1/2	1	1	2	1	4	323	423	513	523	52
221411	3	2.92	1	12	SAN	AB	EXT		1/4	1	1	0	0	1	334	434	51	51	53
221421	3	2.92	1	12	SAN	AB	EXT		3/8	1	1	0	0	1	334	434	51	51	51
221431	5	1.46	1	12	SAN	AB	EXT		1/2	1	1	1	0	2	334	434	524	51	53
221432	5	1.46	2	12	SAN	AB	EXT		1/2	1	1	1	0	2	334	434	534	51	56
221441	5	1.33	1	12	SAN	AB	EXT		5/8	1	1	1	0	2	334	434	524	51	52
221451	5	1.46	1	12	SAN	AB	EXT		3/4	1	1	1	0	2	334	434	524	51	51
222111	3	4.17	1	8	SAN	AB	INT		1/4	1	1	0	0	1	331	431	51	51	63
222121	3	4.17	1	8	SAN	AB	INT		3/8	1	1	0	0	1	331	431	51	51	61
222131	5	2.08	1	8	SAN	AB	INT		1/2	1	1	1	0	2	331	431	621	51	63
222141	5	2.08	1	8	SAN	AB	INT		5/8	1	1	1	0	2	331	431	621	51	62
222151	5	2.08	1	8	SAN	AB	INT		3/4	1	1	1	0	2	331	431	621	51	61
222161	7	1.46	1	8	SAN	AB	INT	1	0/0	1	1	1	1	3	331	431	621	611	62
222171	7	1.46	1	8	SAN	AB	INT	1	1/8	1	1	2	0	3	331	431	621	51	61
222181	7	1.46	1	8	SAN	AB	INT	1	1/4	1	1	2	0	3	321	421	621	51	61
222191	9	0.83	1	8	SAN	AB	INT	1	1/2	1	1	2	1	4	321	421	611	621	62
222331	5	1.75	1	10	SAN	AB	INT		1/2	1	1	1	0	2	333	433	523	51	53
222351	5	1.75	1	10	SAN	AB	INT		3/4	1	1	1	0	2	333	433	523	51	51
222431	5	1.46	1	12	SAN	AB	INT		1/2	1	1	1	0	2	334	434	624	51	63
222451	5	1.46	1	12	SAN	AB	INT		3/4	1	1	1	0	2	334	434	624	51	61
231111	3	4.17	1	8	SAN	AC	EXT		1/4	1	1	0	0	1	331	531	51	51	53
231112	3	4.17	2	8	SAN	AC	EXT		1/4	1	1	0	0	1	341	541	51	51	55
231121	3	4.17	1	8	SAN	AC	EXT		3/8	1	1	0	0	1	331	531	51	51	51
231132	5	2.08	2	8	SAN	AC	EXT		1/2	1	1	1	0	2	331	531	521	51	53
231134	5	2.09	4	8	SAN	AC	EXT		1/2	1	1	1	0	2	331	531	531	51	56
231141	5	2.08	1	8	SAN	AC	EXT		5/8	1	1	1	0	2	331	531	521	51	52
231151	5	2.08	1	8	SAN	AC	EXT		3/4	1	1	1	0	2	331	531	521	51	51
231161	7	1.46	1	8	SAN	AC	EXT	1	0/0	1	1	1	1	3	331	531	521	511	52
231171	7	1.46	1	9	SAN	AC	EXT	1	1/8	1	1	2	0	3	331	531	521	51	51
231181	7	1.46	1	8	SAN	AC	EXT	1	1/4	1	1	2	0	3	321	521	521	51	51
231191	9	0.83	1	9	SAN	AC	EXT	1	1/2	1	1	2	1	4	321	521	511	521	52
231211	3	4.17	1	9	SAN	AC	EXT		1/4	1	1	0	0	1	332	532	51	51	53
231221	3	4.17	1	9	SAN	AC	EXT		3/8	1	1	0	0	1	332	532	51	51	51
231231	5	2.08	1	9	SAN	AC	EXT		1/2	1	1	1	0	2	332	532	522	51	53
231232	5	2.08	2	9	SAN	AC	EXT		1/2	1	1	1	0	2	332	532	532	51	56
231241	5	2.08	1	9	SAN	AC	EXT		5/8	1	1	1	0	2	332	532	522	51	52
231251	5	2.08	1	9	SAN	AC	EXT		3/4	1	1	1	0	2	332	532	522	51	51
231262	7	1.46	1	9	SAN	AC	EXT	1	0/0	1	1	1	1	3	332	532	522	512	52
231271	7	1.46	1	9	SAN	AC	EXT	1	1/8	1	1	2	0	3	332	532	522	51	51
231281	7	1.46	1	9	SAN	AC	EXT	1	1/4	1	1	2	0	3	322	522	522	51	51
231291	9	0.83	1	9	SAN	AC	EXT	1	1/2	1	1	2	1	4	322	522	512	522	52
231311	3	3.50	1	10	SAN	AC	EXT		1/4	1	1	0	0	1	333	533	51	51	53
231321	3	3.50	1	10	SAN	AC	EXT		3/8	1	1	0	0	1	333	533	51	51	51
231331	5	1.75	1	10	SAN	AC	EXT		1/2	1	1	1	0	2	333	533	523	51	53
231332	5	1.75	2	10	SAN	AC	EXT		1/2	1	1	1	0	2	333	533	533	51	56
231341	5	1.75	1	10	SAN	AC	EXT		5/8	1	1	1	0	2	333	533	523	51	52
231351	5	1.75	1	10	SAN	AC	EXT		3/4	1	1	1	0	2	333	533	523	51	51
231361	7	1.25	1	10	SAN	AC	EXT	1	0/0	1	1	1	1	3	333	533	523	513	52
231371	7	1.25	1	10	SAN	AC	EXT	1	1/8	1	1	2	0	3	333	533	523	51	51
231381	7	1.25	1	10	SAN	AC	EXT	1	1/4	1	1	2	0	3	323	523	523	51	51
231391	9	0.83	1	10	SAN	AC	EXT	1	1/2	1	1	2	1	4	323	523	513	523	52
231411	3	2.92	1	12	SAN	AC	EXT		1/4	1	1	0	0	1	334	534	51	51	53
231421	3	2.92	1	12	SAN	AC	EXT		3/8	1	1	0	0	1	334	534	51	51	51
231431	5	1.46	1	12	SAN	AC	EXT		1/2	1	1	1	0	2	334	534	524	51	53
231432	5	1.46	2	12	SAN	AC	EXT		1/2	1	1	1	0	2	334	534	534	51	56
231441	5	1.33	1	12	SAN	AC	EXT		5/8	1	1	1	0	2	334	534	524	51	52
231451	5	1.46	1	12	SAN	AC	EXT		3/4	1	1	1	0	2	334	534	524	51	51
231461	7	1.04	1	12	SAN	AC	EXT	1	0/0	1	1	1	1	3	334	534	524	514	52
231471	7	1.04	1	12	SAN	AC	EXT	1	1/8	1	1	2	0	3	334	534	524	51	51
231482	7	1.04	2	12	SAN	AC	EXT	1	1/4	1	1	2	0	3	324	524	524	51	51
231491	9	0.79	1	12	SAN	AC	EXT	1	1/2	1	1	2	1	4	324	524	514	524	52
242111	3	4.17	1	8	SAN	AD	INT		1/4	1	1	0	0	1	331	631	51	51	63
242112	3	4.17	2	8	SAN	AD	INT		1/4	1	1	0	0	1	341	641	51	51	62
242121	3	4.17	1	8	SAN	AD	INT		3/8	1	1	0	0	1	331	631	51	51	61
242131	5	2.08	1	8	SAN	AD	INT		1/2	1	1	1	0	2	331	631	621	51	63
242132	5	2.08	2	8	SAN	AD	INT		1/2	1	1	1	0	2	331	631	631	51	66
242133	5	2.08	3	8	SAN	AD	INT		1/2	1	1	1	0	2	331	631	631	51	67
242141	5	2.08	1	8	SAN	AD	INT		5/8	1	1	1	0	2	331	631	621	51	62
242151	5	2.08	1	8	SAN	AD	INT		3/4	1	1	1	0	2	331	631	621	51	61
242161	7	1.46	1	8	SAN	AD	INT	1	0/0	1	1	1	1	3	331	631	621	611	62
242171	7	1.46	1	8	SAN	AD	INT	1	1/8	1	1	2	0	3	331	631	621	51	61
242181	7	1.46	1	8	SAN	AD	INT	1	1/4	1	1	2	0	3	321	621	621	51	61
242191	9	0.83	1	8	SAN	AD	INT	1	1/2	1	1	2	1	4	321	621	611	621	62
242311	3	3.50	1	10	SAN	AD	INT		1/4	1	1	0	0	1	333	633	51	51	63
242321	3	3.50	1	10	SAN	AD	INT		3/8	1	1	0	0	1	333	633	51	51	61

PANEL.DAT cont'd

242331	5	1.75	1	10	SAN	AD	INT		1/2	1	1	1	0	2	51	51	51	51	51
242332	5	1.75	2	10	SAN	AD	INT		1/2	1	1	1	0	2	333	633	623	51	63
242333	5	1.75	3	10	SAN	AD	INT		1/2	1	1	1	0	2	333	633	633	51	66
242334	5	1.75	4	10	SAN	AD	INT		1/2	1	1	1	0	2	333	633	633	51	67
242351	5	1.75	1	10	SAN	AD	INT		3/4	1	1	1	0	2	333	633	623	51	61
242411	3	2.92	1	12	SAN	AD	INT		1/4	1	1	0	0	1	334	634	51	51	63
242421	3	2.92	1	12	SAN	AD	INT		3/8	1	1	0	0	1	334	634	51	51	61
242431	5	1.46	1	12	SAN	AD	INT		1/2	1	1	1	0	2	334	634	624	51	63
242432	5	1.46	2	12	SAN	AD	INT		1/2	1	1	1	0	2	334	634	634	51	66
242433	5	1.46	3	12	SAN	AD	INT		1/2	1	1	1	0	2	334	634	634	51	67
242441	5	1.46	1	12	SAN	AD	INT		5/8	1	1	1	0	2	334	634	614	51	62
242451	5	1.46	1	12	SAN	AD	INT		3/4	1	1	1	0	2	334	634	624	51	61
211111	3	4.17	1	8	SAN	AA	EXT		1/4	1	1	0	0	1	331	331	51	51	53
211141	5	2.08	1	8	SAN	AA	EXT		5/8	1	1	1	0	2	331	231	521	51	52
211151	5	2.08	1	8	SAN	AA	EXT		3/4	1	1	1	0	2	331	331	521	51	51
221171	7	1.46	1	8	SAN	AB	EXT	1	1/8	1	1	2	0	3	331	431	521	51	51
221191	9	0.83	1	8	SAN	AB	EXT	1	1/2	1	1	2	1	4	331	421	511	521	52
221211	3	4.17	1	9	SAN	AB	EXT		1/4	1	1	0	0	1	332	432	51	51	53
221251	5	2.08	1	9	SAN	AB	EXT		3/4	1	1	1	0	2	332	432	522	51	51
222241	5	2.08	1	9	SAN	AB	INT		5/8	1	1	1	0	2	332	432	622	51	62
211141	5	2.08	1	8	SAN	BB	EXT		5/8	1	1	1	0	2	431	431	521	51	52
211151	5	2.08	1	8	SAN	BB	EXT		3/4	1	1	1	0	2	431	431	521	51	51
311111	3	4.17	1	8	MAR	AA	EXT		1/4	1	1	0	0	1	831	831	51	51	73
311121	5	2.08	1	8	MAR	AA	EXT		3/8	1	1	1	0	2	831	831	731	51	74
311131	5	2.08	1	8	MAR	AA	EXT		1/2	1	1	1	0	2	231	831	721	51	73
311141	7	1.46	1	8	MAR	AA	EXT		5/8	1	1	2	0	3	831	831	731	51	73
311151	7	1.46	1	8	MAR	AA	EXT		3/4	1	1	2	0	3	831	831	721	51	73
311161	9	0.83	1	8	MAR	AA	EXT	1	0/0	1	1	3	0	4	831	831	721	51	73
311311	3	3.50	1	10	MAR	AA	EXT		1/4	1	1	0	0	1	833	833	51	51	73
311321	5	1.75	1	10	MAR	AA	EXT		3/8	1	1	1	0	2	833	833	733	51	74
311331	5	1.75	1	10	MAR	AA	EXT		1/2	1	1	1	0	2	833	833	723	51	73
311341	7	1.25	1	10	MAR	AA	EXT		5/8	1	1	2	0	3	833	833	733	51	73
311351	7	1.25	1	10	MAR	AA	EXT		3/4	1	1	2	0	3	833	833	723	51	73
311361	9	0.83	1	10	MAR	AA	EXT	1	0/0	1	1	3	0	4	833	833	723	51	73
311412	3	2.92	2	12	MAR	AA	EXT		1/4	1	1	0	0	1	834	834	51	51	73
311421	5	1.46	1	12	MAR	AA	EXT		3/8	1	1	1	0	2	834	834	734	51	74
311431	5	1.46	1	12	MAR	AA	EXT		1/2	1	1	1	0	2	834	834	724	51	73
311441	7	1.04	1	12	MAR	AA	EXT		5/8	1	1	2	0	3	834	834	734	51	73
311451	7	1.04	1	12	MAR	AA	EXT		3/4	1	1	2	0	3	834	834	724	51	73
311461	9	0.79	1	12	MAR	AA	EXT	1	0/0	1	1	3	0	4	834	834	724	51	73
321111	3	4.17	1	8	MAR	AB	EXT		1/4	1	1	0	0	1	831	431	51	51	73
321121	5	2.08	1	8	MAR	AB	EXT		3/8	1	1	1	0	2	831	431	731	51	74
321131	5	2.08	1	8	MAR	AB	EXT		1/2	1	1	1	0	2	831	431	721	51	73
321141	7	1.46	1	8	MAR	AB	EXT		5/8	1	1	2	0	3	831	431	731	51	73
321151	7	1.46	1	8	MAR	AB	EXT		3/4	1	1	2	0	3	831	431	721	51	73
321161	9	0.83	1	8	MAR	AB	EXT	1	0/0	1	1	3	0	4	831	431	721	51	73
321241	7	1.46	1	9	MAR	AB	EXT		5/8	1	1	2	0	3	832	432	732	51	73
321251	7	1.46	1	9	MAR	AB	EXT		3/4	1	1	2	0	3	832	432	722	51	73
321311	3	3.50	1	10	MAR	AB	EXT		1/4	1	1	0	0	1	833	433	51	51	73
321321	5	1.75	1	10	MAR	AB	EXT		3/8	1	1	1	0	2	833	433	733	51	74
321331	5	1.75	1	10	MAR	AB	EXT		1/2	1	1	1	0	2	833	433	723	51	73
321341	7	1.17	1	10	MAR	AB	EXT		5/8	1	1	2	0	3	833	433	733	51	73
321351	7	1.17	1	10	MAR	AB	EXT		3/4	1	1	2	0	3	833	433	723	51	73
321361	9	0.83	1	10	MAR	AB	EXT	1	0/0	1	1	3	0	4	833	433	723	51	73
321411	3	2.92	1	12	MAR	AB	EXT		1/4	1	1	0	0	1	834	434	51	51	73
321421	5	1.46	1	12	MAR	AB	EXT		3/8	1	1	1	0	2	834	434	734	51	74
321431	5	1.46	1	12	MAR	AB	EXT		1/2	1	1	1	0	2	834	434	724	51	73
321441	7	1.04	1	12	MAR	AB	EXT		5/8	1	1	2	0	3	834	434	734	51	73
321451	7	1.04	1	12	MAR	AB	EXT		3/4	1	1	2	0	3	834	434	724	51	73
321461	9	0.79	1	12	MAR	AB	EXT	1	0/0	1	1	3	0	4	834	434	724	51	73
411101	3	3.75	1	8	DUR	G1S	EXT		5/16	1	1	0	0	1	431	531	51	51	53
411111	3	3.75	1	8	DUR	G1S	EXT		1/4	1	1	0	0	1	431	531	51	51	54
411121	3	3.75	1	8	DUR	G1S	EXT		3/8	1	1	0	0	1	431	531	51	51	52
411131	5	1.87	1	8	DUR	G1S	EXT		1/2	1	1	1	0	2	431	531	531	51	53
411141	5	1.87	1	8	DUR	G1S	EXT		5/8	1	1	1	0	2	431	531	531	51	52
411151	5	1.87	1	8	DUR	G1S	EXT		3/4	1	1	1	0	2	431	531	511	51	52
411201	3	3.75	1	9	DUR	G1S	EXT		5/16	1	1	0	0	1	432	532	51	51	53
411211	3	3.75	1	9	DUR	G1S	EXT		1/4	1	1	0	0	1	432	532	51	51	54
411221	3	3.75	1	9	DUR	G1S	EXT		3/8	1	1	0	0	1	432	532	51	51	52
411231	5	1.87	1	9	DUR	G1S	EXT		1/2	1	1	1	0	2	432	532	532	51	53
411241	5	1.87	1	9	DUR	G1S	EXT		5/8	1	1	1	0	2	432	532	532	51	52
411251	5	1.87	1	9	DUR	G1S	EXT		3/4	1	1	1	0	2	432	532	512	51	52
411252	5	1.87	2	9	DUR	G1S	EXT		3/4	1	1	1	0	2	432	532	532	51	51
411301	3	3.17	1	10	DUR	G1S	EXT		5/16	1	1	0	0	1	433	533	51	51	53
411311	3	3.17	1	10	DUR	G1S	EXT		1/4	1	1	0	0	1	433	533	51	51	54
411321	3	3.17	1	10	DUR	G1S	EXT		3/8	1	1	0	0	1	433	533	51	51	52
411331	5	1.58	1	10	DUR	G1S	EXT		1/2	1	1	1	0	2	433	533	533	51	53
411341	5	1.58	1	10	DUR	G1S	EXT		5/8	1	1	1	0	2	433	533	533	51	52
411351	5	1.58	1	10	DUR	G1S	EXT		3/4	1	1	1	0	2	433	533	513	51	52
411352	5	1.58	2	10	DUR	G1S	EXT		3/4	1	1	1	0	2	433	533	533	51	51
411401	3	2.67	1	12	DUR	G1S	EXT		5/16	1	1	0	0	1	434	534	51	51	53
411411	3	2.67	1	12	DUR	G1S	EXT		1/4	1	1	0	0	1	434	534	51	51	54

PANEL.DAT cont'd

411421	3	2.67	1	12	DUR	G1S	EXT		3/8	1	1	0	0	1	434	534	51	51	52
411431	5	1.33	1	12	DUR	G1S	EXT		1/2	1	1	1	0	2	434	534	534	51	53
411441	5	1.33	1	12	DUR	G1S	EXT		5/8	1	1	1	0	2	434	534	534	51	52
411451	5	1.33	1	12	DUR	G1S	EXT		3/4	1	1	1	0	2	434	534	514	51	52
411452	5	1.33	2	12	DUR	G1S	EXT		3/4	1	1	1	0	2	434	534	534	51	51
421101	3	1.87	1	8	DUR	G2S	EXT		5/16	1	1	0	0	1	431	431	51	51	53
421111	3	1.87	1	8	DUR	G2S	EXT		1/4	1	1	0	0	1	431	431	51	51	54
421121	3	1.87	1	8	DUR	G2S	EXT		3/8	1	1	0	0	1	431	431	51	51	52
421131	5	1.87	1	8	DUR	G2S	EXT		1/2	1	1	1	0	2	431	431	531	51	53
421141	5	1.87	1	8	DUR	G2S	EXT		5/8	1	1	1	0	2	431	431	531	51	52
421151	5	1.87	1	8	DUR	G2S	EXT		3/4	1	1	1	0	2	431	431	511	51	52
421161	7	1.46	1	8	DUR	G2S	EXT	1	0/0	1	1	2	0	3	431	431	521	51	52
421201	3	1.87	1	9	DUR	G2S	EXT		5/16	1	1	0	0	1	432	432	51	51	53
421221	3	1.87	1	9	DUR	G2S	EXT		3/8	1	1	0	0	1	432	432	51	51	52
421231	5	1.87	1	9	DUR	G2S	EXT		1/2	1	1	1	0	2	432	432	532	51	53
421241	5	1.87	1	9	DUR	G2S	EXT		5/8	1	1	1	0	2	432	432	532	51	52
421251	5	1.87	1	9	DUR	G2S	EXT		3/4	1	1	1	0	2	432	432	512	51	52
421252	5	1.87	2	9	DUR	G2S	EXT		3/4	1	1	1	0	2	432	432	532	51	51
421301	3	1.58	1	10	DUR	G2S	EXT		5/16	1	1	0	0	1	433	433	51	51	53
421311	3	1.58	1	10	DUR	G2S	EXT		1/4	1	1	0	0	1	433	433	51	51	54
421321	3	1.58	1	10	DUR	G2S	EXT		3/8	1	1	0	0	1	433	433	51	51	52
421331	5	1.58	1	10	DUR	G2S	EXT		1/2	1	1	1	0	2	433	433	533	51	53
421341	5	1.58	1	10	DUR	G2S	EXT		5/8	1	1	1	0	2	433	433	533	52	51
421351	5	1.58	1	10	DUR	G2S	EXT		3/4	1	1	1	0	2	433	433	513	51	52
421352	5	1.58	2	10	DUR	G2S	EXT		3/4	1	1	1	0	2	433	433	533	51	51
421361	7	1.13	1	10	DUR	G2S	EXT	1	0/0	1	1	2	0	3	433	433	523	51	52
421401	3	1.33	1	12	DUR	G2S	EXT		5/16	1	1	0	0	1	434	434	51	51	53
421411	3	1.33	1	12	DUR	G2S	EXT		1/4	1	1	0	0	1	434	434	51	51	54
421421	3	1.33	1	12	DUR	G2S	EXT		3/8	1	1	0	0	1	434	434	51	51	52
421431	5	1.33	1	12	DUR	G2S	EXT		1/2	1	1	1	0	2	434	434	534	51	53
421441	5	1.33	1	12	DUR	G2S	EXT		5/8	1	1	1	0	2	434	434	534	51	52
421451	5	1.33	1	12	DUR	G2S	EXT		3/4	1	1	1	0	2	434	434	514	51	52
421452	5	1.33	2	12	DUR	G2S	EXT		3/4	1	1	1	0	2	434	434	534	51	51
421461	7	1.04	1	12	DUR	G2S	EXT	1	0/0	1	1	2	0	3	434	434	524	51	52
431151	7	1.46	1	8	DUR	G2S-CCTR	EXT		3/4	1	1	2	0	3	841	441	551	51	53
431152	7	1.46	2	8	DUR	G2S-CCTR	EXT		3/4	1	1	2	0	3	831	431	571	51	53
431153	7	1.46	3	8	DUR	G2S-CCTR	EXT		3/4	1	1	2	0	3	831	431	531	51	57
431161	9	0.83	1	8	DUR	G2S-CCTR	EXT	1	0/0	1	1	2	1	4	431	431	521	531	53
431451	7	1.04	1	12	DUR	G2S-CCTR	EXT		3/4	1	1	2	0	3	434	434	524	51	53
411161	7	1.46	1	8	DUR	G1S	EXT	1	0/0	1	1	2	0	3	431	531	521	51	52
441151	7	1.46	1	8	DUR	G2S-SCTR	EXT		3/4	1	1	2	0	3	841	441	751	51	53
441152	7	1.46	2	8	DUR	G2S-SCTR	EXT		3/4	1	1	2	0	3	831	431	771	51	53
441153	7	1.46	3	8	DUR	G2S-SCTR	EXT		3/4	1	1	2	0	3	831	431	731	51	57
441161	9	0.83	1	8	DUR	G2S-SCTR	EXT	1	0/0	1	1	2	1	4	431	431	721	731	53
451141	5	1.87	1	8	DUR	CON G1S	EXT		5/8	1	1	1	0	2	431	531	531	51	52
451151	5	1.87	1	8	DUR	CON G1S	EXT		3/4	1	1	1	0	2	431	531	511	51	52
461141	5	1.87	1	8	DUR	CON G2S	EXT		5/8	1	1	1	0	2	431	431	531	51	52
461151	5	1.87	1	8	DUR	CON G2S	EXT		3/4	1	1	1	0	2	431	431	511	51	52
552101	3	4.17	1	8	SHE	CD-PTS	INT		5/16	1	1	0	0	1	731	631	51	51	63
552121	3	4.17	1	8	SHE	CD-PTS	INT		3/8	1	1	0	0	1	731	631	51	51	62
552131	4	2.08	1	8	SHE	CD-PTS	INT		1/2	1	1	0	0	2	761	661	51	51	66
552132	5	2.08	2	8	SHE	CD-PTS	INT		1/2	1	1	1	0	2	731	631	631	51	63
552133	5	2.08	3	8	SHE	CD-PTS	INT		1/2	1	1	1	0	2	731	631	621	51	64
552141	5	2.08	1	8	SHE	CD-PTS	INT		5/8	1	1	1	0	2	731	631	651	51	65
552142	5	2.08	2	8	SHE	CD-PTS	INT		5/8	1	1	1	0	2	731	631	621	51	62
552143	5	2.08	3	8	SHE	CD-PTS	INT		5/8	1	1	1	0	2	761	661	661	51	66
552151	5	2.08	1	8	SHE	CD-PTS	INT		3/4	1	1	1	0	2	731	631	611	51	62
552152	5	2.08	2	8	SHE	CD-PTS	INT		3/4	1	1	1	0	2	721	621	631	51	62
552153	5	2.08	3	8	SHE	CD-PTS	INT		3/4	1	1	1	0	2	721	621	621	51	66
552154	5	2.08	4	8	SHE	CD-PTS	INT		3/4	1	1	0	0	3	761	661	51	51	62
552161	7	1.46	1	8	SHE	CD-PTS	INT	1	0/0	1	1	2	0	3	731	631	621	51	62
552162	7	1.46	2	8	SHE	CD-PTS	INT	1	0/0	1	1	2	0	3	721	621	631	51	62
552231	5	2.08	1	9	SHE	CD-PTS	INT		1/2	1	1	1	0	2	732	632	632	51	63
552341	5	1.75	1	10	SHE	CD-PTS	INT		5/8	1	1	1	0	2	733	633	633	51	62
552351	5	1.75	1	10	SHE	CD-PTS	INT		3/4	1	1	1	0	2	733	633	613	51	62
552352	5	1.75	2	10	SHE	CD-PTS	INT		3/4	1	1	1	0	2	723	623	633	51	62
552353	5	1.75	3	10	SHE	CD-PTS	INT		3/4	1	1	1	0	2	723	623	623	51	66
552354	5	1.75	4	10	SHE	CD-PTS	INT		3/4	1	1	1	0	2	723	623	623	51	66
552441	5	1.33	1	12	SHE	CD-PTS	INT		5/8	1	1	1	0	2	734	634	634	51	62
552451	5	1.46	1	12	SHE	CD-PTS	INT		3/4	1	1	1	0	2	734	634	614	51	62
561171	7	1.46	1	8	SHE	2-4-1	INT	1	1/8	1	1	2	0	3	721	621	621	51	62
561172	7	1.46	2	8	SHE	2-4-1	INT	1	1/8	1	1	1	1	3	721	621	611	621	65
561173	7	1.46	3	8	SHE	2-4-1	INT	1	1/8	1	1	2	1	2	731	631	611	52	62
561174	7	1.46	4	8	SHE	2-4-1	INT	1	1/8	1	1	2	1	2	731	631	[illegible]	51	[illegible]
512101	3	4.17	1	8	SHE	CDX	INT		5/16	1	1	0	0	1	531	631	51	51	63
512102	3	4.17	2	8	SHE	CDX	INT		5/16	1	1	0	0	1	541	641	51	51	62
512103	3	4.17	3	8	SHE	CDX	INT		5/16	1	1	0	0	1	541	641	51	51	65
512121	2	4.17	1	8	SHE	CDX	INT		3/8	1	1	0	0	1	561	661	51	51	66
512122	3	4.17	2	8	SHE	CDX	INT		3/8	1	1	0	0	1	531	631	51	51	62
512131	5	1.87	1	8	SHE	CDX	INT		1/2	1	1	1	0	2	531	631	631	51	63
512132	5	1.87	2	8	SHE	CDX	INT		1/2	1	1	1	0	2	531	631	621	51	64
512133	5	1.87	3	8	SHE	CDX	INT		1/2	1	1	1	0	2	531	631	641	51	66

512134	5	1.37	4	8	SHE	CDX	INT		1/2	1	1	1	0	2	531	631	641	51	67
512141	4	1.37	1	8	SHE	CDX	INT		5/8	1	1	0	0	2	521	621	51	51	65
512142	5	1.37	2	8	SHE	CDX	INT		5/8	1	1	1	0	2	531	631	631	51	62
512143	5	1.37	3	8	SHE	CDX	INT		5/8	1	1	1	0	2	531	631	651	51	65
512144	5	1.87	4	8	SHE	CDX	INT		5/8	1	1	1	0	2	561	661	661	51	66
512151	5	2.08	1	8	SHE	CDX	INT		3/4	1	1	1	0	2	531	631	611	51	62
512152	5	2.08	2	8	SHE	CDX	INT		3/4	1	1	1	0	2	521	621	611	51	63
512153	5	2.08	3	8	SHE	CDX	INT		3/4	1	1	1	0	2	521	621	621	51	66
512154	5	2.08	4	8	SHE	CDX	INT		3/4	1	1	0	0	3	561	661	51	51	62
512161	7	1.46	1	8	SHE	CDX	INT	1	0/0	1	1	2	0	3	531	631	621	51	62
512162	7	1.46	2	8	SHE	CDX	INT	1	0/0	1	1	2	0	3	521	621	631	51	62
512181	7	1.46	1	8	SHE	CDX	INT	1	1/4	1	1	2	0	3	521	521	611	51	62
512191	9	0.83	1	8	SHE	CDX	INT	1	1/2	1	1	2	1	4	521	621	621	611	62
512223	3	4.17	3	9	SHE	CDX	INT		3/8	1	1	0	0	1	532	632	51	51	62
512251	5	2.08	1	9	SHE	CDX	INT		3/4	1	1	1	0	2	532	632	612	51	62
512252	5	2.08	2	9	SHE	CDX	INT		3/4	1	1	1	0	2	522	622	612	51	63
512253	5	2.08	3	9	SHE	CDX	INT		3/4	1	1	1	0	2	522	622	622	51	66
512301	3	3.50	1	10	SHE	CDX	INT		5/16	1	1	0	0	1	533	633	51	51	63
512321	3	3.50	1	10	SHE	CDX	INT		3/8	1	1	0	0	1	533	633	51	51	62
512331	4	1.75	1	10	SHE	CDX	INT		1/2	1	1	0	1	1	533	633	51	65	62
512332	5	1.75	2	10	SHE	CDX	INT		1/2	1	1	1	0	2	533	633	633	51	63
512333	5	1.75	3	10	SHE	CDX	INT		1/2	1	1	1	0	2	533	633	623	51	64
512341	4	1.75	1	10	SHE	CDX	INT		5/8	1	1	0	0	2	523	623	51	51	65
512342	5	1.75	2	10	SHE	CDX	INT		5/8	1	1	1	0	2	533	633	633	51	62
512351	5	1.75	1	10	SHE	CDX	INT		3/4	1	1	1	0	2	533	633	613	51	62
512352	5	1.75	2	10	SHE	CDX	INT		3/4	1	1	1	0	2	523	623	613	51	63
512353	5	1.75	3	10	SHE	CDX	INT		3/4	1	1	1	0	2	523	623	623	51	66
512401	3	2.92	1	12	SHE	CDX	INT		5/16	1	1	0	0	1	534	634	51	51	63
512421	3	2.92	1	12	SHE	CDX	INT		3/8	1	1	0	0	1	534	634	51	51	62
512431	4	1.46	1	12	SHE	CDX	INT		1/2	1	1	0	1	1	534	634	51	65	62
512432	5	1.46	2	12	SHE	CDX	INT		1/2	1	1	1	0	2	534	634	634	51	63
512433	5	1.46	3	12	SHE	CDX	INT		1/2	1	1	1	0	2	534	634	624	51	64
512441	4	1.46	1	12	SHE	CDX	INT		5/8	1	1	0	0	2	524	624	51	51	65
512442	5	1.46	2	12	SHE	CDX	INT		5/8	1	1	1	0	2	534	634	634	51	62
512451	5	1.46	1	12	SHE	CDX	INT		3/4	1	1	1	0	2	534	634	614	51	62
512452	5	1.46	2	12	SHE	CDX	INT		3/4	1	1	1	0	2	524	624	614	51	63
512453	5	1.46	3	12	SHE	CDX	INT		3/4	1	1	1	0	2	524	624	624	51	66
521141	7	1.46	1	8	SHE	CON G1S	EXT		5/8	1	1	2	0	3	441	541	541	51	53
532101	3	4.17	1	8	SHE	AA	INT		5/16	1	1	0	0	1	631	631	51	51	64
532102	3	4.17	2	8	SHE	AA	INT		5/16	1	1	0	0	1	631	641	51	51	63
541101	3	4.17	1	8	SHE	CC-PTS	EXT		5/16	1	1	0	0	1	731	531	51	51	53
541122	3	4.17	2	8	SHE	CC-PTS	EXT		3/8	1	1	0	0	1	731	531	51	51	52
541131	4	2.08	1	8	SHE	CC-PTS	EXT		1/2	1	1	0	0	2	761	561	51	51	56
541132	5	2.08	2	8	SHE	CC-PTS	EXT		1/2	1	1	1	0	2	731	531	531	51	53
541141	5	2.08	1	9	SHE	CC-PTS	EXT		5/8	1	1	1	0	2	731	531	531	51	52
541142	5	2.08	2	8	SHE	CC-PTS	EXT		5/8	1	1	1	0	2	761	561	561	51	56
541151	5	2.08	1	8	SHE	CC-PTS	EXT		3/4	1	1	1	0	2	731	531	511	51	52
541161	7	1.46	1	8	SHE	CC-PTS	EXT	1	0/0	1	1	2	0	3	731	531	521	51	52
541201	3	4.17	1	9	SHE	CC-PTS	EXT		5/16	1	1	0	0	1	732	532	51	51	53
541221	3	4.17	1	9	SHE	CC-PTS	EXT		3/8	1	1	0	0	1	732	532	51	51	52
541231	5	2.08	1	9	SHE	CC-PTS	EXT		1/2	1	1	1	0	2	732	532	532	51	53
541241	5	2.08	1	9	SHE	CC-PTS	EXT		5/8	1	1	1	0	2	732	532	532	51	52
541251	5	2.08	1	9	SHE	CC-PTS	EXT		3/4	1	1	1	0	2	732	532	512	51	52
541261	7	1.46	1	9	SHE	CC-PTS	EXT	1	0/0	1	1	2	0	3	732	532	522	51	52
541301	3	3.50	1	10	SHE	CC-PTS	EXT		5/16	1	1	0	0	1	733	533	51	51	56
541321	3	3.50	1	10	SHE	CC-PTS	EXT		3/8	1	1	0	0	1	733	533	51	51	52
541331	5	1.75	1	10	SHE	CC-PTS	EXT		1/2	1	1	1	0	2	733	533	533	51	53
541341	5	1.75	1	10	SHE	CC-PTS	EXT		5/8	1	1	1	0	2	733	533	533	51	52
541352	5	1.75	2	10	SHE	CC-PTS	EXT		3/4	1	1	1	0	2	733	533	513	51	52
541361	7	1.25	1	10	SHE	CC-PTS	EXT	1	0/0	1	1	2	0	3	733	533	523	51	52
541401	3	2.92	1	12	SHE	CC-PTS	EXT		5/16	1	1	0	0	1	734	534	51	51	53
541422	3	2.92	2	12	SHE	CC-PTS	EXT		3/8	1	1	0	0	1	734	534	51	51	52
541431	5	1.39	1	12	SHE	CC-PTS	EXT		1/2	1	1	1	0	2	734	534	534	51	53
541441	5	1.46	1	12	SHE	CC-PTS	EXT		5/8	1	1	1	0	2	734	534	534	51	52
541451	5	1.46	1	12	SHE	CC-PTS	EXT		3/4	1	1	1	0	2	734	534	514	51	52
511121	3	4.17	1	8	SHE	CCX	EXT		3/8	1	1	0	0	1	531	531	51	51	52
511131	5	2.08	1	8	SHE	CCX	EXT		1/2	1	1	1	0	2	531	531	531	51	53
511141	5	2.08	1	8	SHE	CCX	EXT		5/8	1	1	1	0	2	531	531	531	51	52
511151	5	2.08	1	8	SHE	CCX	EXT		3/4	1	1	1	0	2	631	631	511	51	52
511161	7	1.46	1	8	SHE	CCX	EXT	1	0/0	1	1	2	0	3	531	531	521	51	52
511171	7	1.46	1	8	SHE	CCX	EXT	1	1/8	1	1	2	0	3	521	521	521	51	52
511191	9	0.83	1	8	SHE	CCX	EXT	1	1/2	1	1	2	1	4	521	521	521	511	52
511251	5	1.87	1	9	SHE	CCX	EXT		3/4	1	1	1	0	2	532	532	512	51	52
521121	3	4.17	1	8	SHE	AC	EXT		3/8	1	1	0	0	1	731	731	51	51	52
521151	5	2.08	1	8	SHE	AC	EXT		3/4	1	1	1	0	2	731	731	511	51	52
521161	7	1.46	1	8	SHE	AC	EXT	1	0/0	1	1	2	0	3	731	731	521	51	52
521171	7	1.46	1	8	SHE	AC	EXT	1	1/8	1	1	2	0	3	721	721	521	51	52
521241	5	2.08	1	9	SHE	AC	EXT		5/8	1	1	1	0	2	732	732	532	51	52
521271	7	14.58	1	9	SHE	AC	EXT	1	1/8	1	1	2	0	3	722	722	522	51	52
521291	9	0.83	1	9	SHE	AC	EXT	1	1/2	1	1	2	1	4	722	722	522	512	52
521331	5	1.75	1	10	SHE	AC	EXT		1/2	1	1	1	0	2	733	733	533	51	53
521341	5	1.75	1	10	SHE	AC	EXT		5/8	1	1	1	0	2	733	733	533	51	52

PANEL.DAT cont'd

521351	5	1.75	1	10	SHE	AC	EXT	3/4	1	1	1	0	2	733	733	513	51	52
512241	5	2.08	1	9	SHE	CDX	INT	5/8	1	1	1	0	2	532	632	632	51	62
512261	7	1.46	1	9	SHE	CDX	INT	1 0/0	1	1	1	0	3	532	632	522	51	52
512231	5	2.08	1	9	SHE	CDX	INT	1/2	1	1	1	0	2	532	632	632	51	63
552321	3	3.50	1	10	SHE	CD-PTS	INT	3/8	1	1	0	0	1	733	633	51	51	62
552401	3	2.92	1	12	SHE	CD-PTS	INT	5/16	1	1	0	0	1	534	634	51	51	63
552421	3	2.92	1	12	SHE	CD-PTS	INT	3/8	1	1	0	0	1	734	634	51	51	62
552301	3	3.50	1	10	SHE	CD-PTS	INT	5/16	1	1	0	0	1	533	633	51	51	63

PATDEV.DAT Patcher Development & Production rate File

veneer identification code	A-grade	B-grade	CP-grade	C-grade	D-grade	M-grade	ECLR-grade	DCLR-grade	Production rate
9211	0.740	0.149	0.046	0.050	0.000	0.000	0.000	0.000	0.007
9212	0.736	0.138	0.046	0.050	0.000	0.000	0.000	0.000	0.007
9213	0.703	0.132	0.044	0.048	0.000	0.000	0.000	0.000	0.007
9221	0.740	0.185	0.000	0.050	0.000	0.000	0.000	0.000	0.007
9222	0.740	0.185	0.000	0.050	0.000	0.000	0.000	0.000	0.007
9223	0.703	0.132	0.044	0.048	0.000	0.000	0.000	0.000	0.007
9231	0.735	0.138	0.045	0.245	0.000	0.000	0.000	0.000	0.006
9232	0.736	0.138	0.046	0.150	0.000	0.000	0.000	0.000	0.006
9233	0.703	0.132	0.044	0.048	0.000	0.000	0.000	0.000	0.006
9241	0.740	0.139	0.046	0.050	0.000	0.000	0.000	0.000	0.005
9242	0.736	0.138	0.046	0.050	0.000	0.000	0.000	0.000	0.005
9243	0.703	0.132	0.044	0.048	0.000	0.000	0.000	0.000	0.005
9311	0.740	0.139	0.046	0.050	0.000	0.000	0.000	0.000	0.005
9312	0.736	0.138	0.046	0.050	0.000	0.000	0.000	0.000	0.005
9313	0.703	0.132	0.044	0.048	0.000	0.000	0.000	0.000	0.005
9321	0.879	0.046	0.000	0.050	0.000	0.000	0.000	0.000	0.004
9322	0.874	0.046	0.000	0.050	0.000	0.000	0.000	0.000	0.004
9323	0.835	0.044	0.000	0.048	0.000	0.000	0.000	0.000	0.004
9331	0.740	0.139	0.046	0.050	0.000	0.000	0.000	0.000	0.004
9332	0.736	0.138	0.046	0.050	0.000	0.000	0.000	0.000	0.004
9333	0.703	0.132	0.044	0.048	0.000	0.000	0.000	0.000	0.004
9341	0.740	0.139	0.046	0.050	0.000	0.000	0.000	0.000	0.004
9342	0.736	0.138	0.046	0.050	0.000	0.000	0.000	0.000	0.004
9343	0.703	0.132	0.044	0.048	0.000	0.000	0.000	0.000	0.004
9411	0.740	0.139	0.046	0.050	0.000	0.000	0.000	0.000	0.004
9412	0.736	0.138	0.046	0.050	0.000	0.000	0.000	0.000	0.004
9413	0.703	0.132	0.044	0.048	0.000	0.000	0.000	0.000	0.004
9421	0.879	0.046	0.000	0.050	0.000	0.000	0.000	0.000	0.004
9422	0.874	0.046	0.000	0.050	0.000	0.000	0.000	0.000	0.004
9423	0.835	0.044	0.000	0.048	0.000	0.000	0.000	0.000	0.004
9431	0.740	0.139	0.046	0.050	0.000	0.000	0.000	0.000	0.004
9432	0.736	0.138	0.046	0.050	0.000	0.000	0.000	0.000	0.004
9433	0.703	0.132	0.044	0.048	0.000	0.000	0.000	0.000	0.004
9441	0.740	0.139	0.046	0.050	0.000	0.000	0.000	0.000	0.003
9442	0.736	0.138	0.046	0.050	0.000	0.000	0.000	0.000	0.003
9443	0.703	0.132	0.044	0.048	0.000	0.000	0.000	0.000	0.003
1111	0.000	0.000	0.000	0.000	0.000	0.500	0.500	0.000	0.009
1121	0.000	0.000	0.000	0.000	0.000	0.500	0.500	0.000	0.009
1131	0.000	0.000	0.000	0.000	0.000	0.500	0.500	0.000	0.008
1141	0.000	0.000	0.000	0.000	0.000	0.500	0.500	0.000	0.006
1211	0.000	0.000	0.000	0.000	0.000	0.500	0.500	0.000	0.007
1221	0.000	0.000	0.000	0.000	0.000	0.500	0.500	0.000	0.007
1231	0.000	0.000	0.000	0.000	0.000	0.500	0.500	0.000	0.006
1241	0.000	0.000	0.000	0.000	0.000	0.500	0.500	0.000	0.005
1311	0.000	0.000	0.000	0.000	0.000	0.500	0.500	0.000	0.005
1321	0.000	0.000	0.000	0.000	0.000	0.500	0.500	0.000	0.004
1331	0.000	0.000	0.000	0.000	0.000	0.500	0.500	0.000	0.004
1341	0.000	0.000	0.000	0.000	0.000	0.500	0.500	0.000	0.004
1411	0.000	0.000	0.000	0.000	0.000	0.500	0.500	0.000	0.004
1421	0.000	0.000	0.000	0.000	0.000	0.500	0.500	0.000	0.004

PATDEV.DAT cont'd

1431	0.000	0.000	0.000	0.000	0.000	0.500	0.500	0.000	0.004
1441	0.000	0.000	0.000	0.000	0.000	0.500	0.500	0.000	0.003
2111	0.000	0.000	0.000	0.000	0.000	0.500	0.000	0.000	0.009
2121	0.000	0.000	0.000	0.000	0.000	0.500	0.000	0.000	0.009
2131	0.000	0.000	0.000	0.000	0.000	0.500	0.000	0.000	0.008
2141	0.000	0.000	0.000	0.000	0.000	0.500	0.000	0.000	0.006
2211	0.000	0.000	0.000	0.000	0.000	0.500	0.000	0.000	0.007
2221	0.000	0.000	0.000	0.000	0.000	0.500	0.000	0.000	0.007
2231	0.000	0.000	0.000	0.000	0.000	0.500	0.000	0.000	0.006
2241	0.000	0.000	0.000	0.000	0.000	0.500	0.000	0.000	0.005
2311	0.000	0.000	0.000	0.000	0.000	0.500	0.000	0.000	0.005
2312	0.000	0.000	0.000	0.000	0.000	0.500	0.000	0.000	0.005
2321	0.000	0.000	0.000	0.000	0.000	0.500	0.000	0.000	0.004
2322	0.000	0.000	0.000	0.000	0.000	0.495	0.000	0.000	0.004
2331	0.000	0.000	0.000	0.000	0.000	0.500	0.000	0.000	0.004
2332	0.000	0.000	0.000	0.000	0.000	0.495	0.000	0.000	0.004
2333	0.950	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.004
2341	0.000	0.000	0.000	0.000	0.000	0.500	0.000	0.000	0.004
2342	0.000	0.000	0.000	0.000	0.000	0.495	0.000	0.000	0.004
2343	0.950	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.004
2411	0.000	0.000	0.000	0.000	0.000	0.500	0.000	0.000	0.004
2421	0.000	0.000	0.000	0.000	0.000	0.500	0.000	0.000	0.004
2431	0.000	0.000	0.000	0.000	0.000	0.500	0.000	0.000	0.004
2441	0.000	0.000	0.000	0.000	0.000	0.500	0.000	0.000	0.003
6211	0.000	0.000	0.000	0.710	0.290	0.000	0.000	0.000	0.007
6212	0.000	0.000	0.000	0.710	0.290	0.000	0.000	0.000	0.007
6213	0.000	0.000	0.000	0.710	0.290	0.000	0.000	0.000	0.007
6221	0.000	0.000	0.000	0.710	0.290	0.000	0.000	0.000	0.007
6222	0.000	0.000	0.000	0.710	0.000	0.000	0.000	0.000	0.007
6223	0.000	0.000	0.000	0.710	0.000	0.000	0.000	0.000	0.007
6231	0.000	0.000	0.000	0.710	0.290	0.000	0.000	0.000	0.006
6232	0.000	0.000	0.000	0.710	0.000	0.000	0.000	0.000	0.006
6233	0.000	0.000	0.000	0.710	0.000	0.000	0.000	0.000	0.006
6241	0.000	0.000	0.000	0.710	0.000	0.000	0.000	0.000	0.005
6242	0.000	0.000	0.000	0.710	0.000	0.000	0.000	0.000	0.005
6243	0.000	0.000	0.000	0.710	0.000	0.000	0.000	0.000	0.005
6311	0.000	0.000	0.000	0.710	0.290	0.000	0.000	0.000	0.005
6312	0.000	0.000	0.000	0.710	0.290	0.000	0.000	0.000	0.005
6313	0.000	0.000	0.000	0.710	0.290	0.000	0.000	0.000	0.005
6321	0.000	0.000	0.000	0.710	0.290	0.000	0.000	0.000	0.004
6322	0.000	0.000	0.000	0.710	0.290	0.000	0.000	0.000	0.004
6323	0.000	0.000	0.000	0.710	0.290	0.000	0.000	0.000	0.004
6331	0.000	0.000	0.000	0.710	0.290	0.000	0.000	0.000	0.004
6332	0.000	0.000	0.000	0.710	0.290	0.000	0.000	0.000	0.004
6333	0.000	0.000	0.000	0.710	0.290	0.000	0.000	0.000	0.004
6341	0.000	0.000	0.000	0.710	0.290	0.000	0.000	0.000	0.004
6342	0.000	0.000	0.000	0.710	0.290	0.000	0.000	0.000	0.004
6343	0.000	0.000	0.000	0.710	0.290	0.000	0.000	0.000	0.004
6411	0.000	0.000	0.000	0.710	0.290	0.000	0.000	0.000	0.004
6412	0.000	0.000	0.000	0.710	0.290	0.000	0.000	0.000	0.004
6413	0.000	0.000	0.000	0.710	0.290	0.000	0.000	0.000	0.004
6421	0.000	0.000	0.000	0.710	0.290	0.000	0.000	0.000	0.004
6422	0.000	0.000	0.000	0.710	0.290	0.000	0.000	0.000	0.004
6423	0.000	0.000	0.000	0.710	0.290	0.000	0.000	0.000	0.004
6431	0.000	0.000	0.000	0.710	0.290	0.000	0.000	0.000	0.004
6432	0.000	0.000	0.000	0.710	0.290	0.000	0.000	0.000	0.004
6433	0.000	0.000	0.000	0.710	0.290	0.000	0.000	0.000	0.004
6441	0.000	0.000	0.000	0.710	0.290	0.000	0.000	0.000	0.003
6442	0.000	0.000	0.000	0.710	0.290	0.000	0.000	0.000	0.003
6443	0.000	0.000	0.000	0.710	0.290	0.000	0.000	0.000	0.003

DRYDEV.DAT Dryer Development & Production Rate File

veneer identification code	ECLR grade	DCLR grade	BP grade	C grade	D grade	X grade	SOL grade	dryer production rate	green veneer grade	thickness	length	width
1411	0.203	0.203	0.203	0.280	0.080	0.030	0.000	0.075	1P	1/12	8	54
1412	0.000	0.000	0.610	0.260	0.100	0.030	0.000	0.078	1P	1/12	8	27
1413	0.000	0.000	0.610	0.260	0.100	0.030	0.000	0.078	1P	1/12	8	RW
1453	0.000	0.000	0.000	0.300	0.200	0.000	0.500	0.096	1P	1/12	4	RW
1311	0.203	0.203	0.204	0.280	0.080	0.030	0.000	0.086	1P	1/10	8	54
1312	0.000	0.317	0.293	0.260	0.100	0.030	0.000	0.081	1P	1/10	8	27
1313	0.000	0.000	0.610	0.260	0.100	0.030	0.000	0.081	1P	1/10	8	RW
1353	0.000	0.000	0.000	0.300	0.200	0.000	0.500	0.110	1P	1/10	4	RW
2411	0.117	0.117	0.271	0.385	0.080	0.030	0.000	0.075	3P	1/12	8	54
2412	0.000	0.000	0.505	0.365	0.100	0.030	0.000	0.078	3P	1/12	8	27
2413	0.000	0.000	0.505	0.365	0.100	0.030	0.000	0.078	3P	1/12	8	RW
2453	0.000	0.000	0.000	0.300	0.200	0.000	0.500	0.096	3P	1/12	4	RW
2311	0.117	0.117	0.271	0.385	0.080	0.030	0.000	0.086	3P	1/10	8	54
2312	0.000	0.090	0.415	0.365	0.100	0.030	0.000	0.081	3P	1/10	8	27
2313	0.000	0.000	0.505	0.365	0.100	0.030	0.000	0.081	3P	1/10	8	RW
2353	0.000	0.000	0.000	0.300	0.200	0.000	0.500	0.110	3P	1/10	4	RW
3411	0.054	0.000	0.236	0.580	0.100	0.030	0.000	0.075	SM	1/12	8	54
3412	0.000	0.000	0.290	0.530	0.150	0.030	0.000	0.078	SM	1/12	8	27
3413	0.000	0.000	0.290	0.530	0.150	0.030	0.000	0.078	SM	1/12	8	RW
3453	0.000	0.000	0.000	0.300	0.200	0.000	0.500	0.096	SM	1/12	4	RW
3311	0.054	0.054	0.182	0.580	0.100	0.030	0.000	0.086	SM	1/10	8	54
3312	0.000	0.840	0.206	0.530	0.150	0.030	0.000	0.081	SM	1/10	8	27
3313	0.000	0.000	0.290	0.530	0.150	0.030	0.000	0.081	SM	1/10	8	RW
3353	0.000	0.000	0.000	0.300	0.200	0.000	0.500	0.110	SM	1/10	4	RW
4311	0.000	0.030	0.330	0.550	0.090	0.000	0.000	0.086	2M	1/10	8	54
4312	0.000	0.000	0.180	0.450	0.370	0.000	0.000	0.081	2M	1/10	8	27
4313	0.000	0.000	0.230	0.570	0.200	0.000	0.000	0.081	2M	1/10	8	RW
4353	0.000	0.000	0.000	0.500	0.100	0.000	0.400	0.110	2M	1/10	4	RW
4211	0.000	0.000	0.360	0.550	0.090	0.000	0.000	0.099	2M	1/6	8	54
4212	0.000	0.000	0.180	0.450	0.370	0.000	0.000	0.088	2M	1/6	8	27
4213	0.000	0.000	0.230	0.570	0.200	0.000	0.000	0.088	2M	1/6	8	RW
4253	0.000	0.000	0.000	0.500	0.100	0.000	0.400	0.111	2M	1/6	4	RW
5311	0.000	0.000	0.000	0.770	0.200	0.030	0.000	0.086	3M	1/10	8	54
5312	0.000	0.000	0.000	0.770	0.200	0.030	0.000	0.081	3M	1/10	8	27
5313	0.000	0.000	0.000	0.770	0.200	0.030	0.000	0.081	3M	1/10	8	RW
5353	0.000	0.000	0.000	0.700	0.200	0.000	0.100	0.110	3M	1/10	4	RW
5211	0.000	0.000	0.000	0.770	0.200	0.030	0.000	0.099	3M	1/6	8	54
5212	0.000	0.000	0.000	0.770	0.200	0.030	0.000	0.088	3M	1/6	8	27
5213	0.000	0.000	0.000	0.770	0.200	0.030	0.000	0.088	3M	1/6	8	RW
5253	0.000	0.000	0.000	0.700	0.200	0.000	0.100	0.111	3M	1/6	4	RW
5112	0.000	0.000	0.000	0.770	0.200	0.030	0.000	0.091	3M	7/32	8	27
5113	0.000	0.000	0.000	0.770	0.200	0.030	0.000	0.091	3M	7/32	8	RW
5153	0.000	0.000	0.000	0.700	0.200	0.000	0.100	0.086	3M	7/32	4	RW
6311	0.000	0.044	0.176	0.390	0.390	0.000	0.000	0.086	PC	1/10	8	54
6312	0.000	0.044	0.176	0.390	0.390	0.000	0.000	0.081	PC	1/10	8	27
6313	0.000	0.000	0.220	0.390	0.390	0.000	0.000	0.081	PC	1/10	8	RW
6353	0.000	0.000	0.000	0.450	0.400	0.000	0.150	0.110	PC	1/10	4	RW
1321	0.202	0.203	0.205	0.280	0.080	0.030	0.000	0.077	1P	1/10	9	54
1322	0.000	0.317	0.293	0.260	0.100	0.030	0.000	0.072	1P	1/10	9	27
1323	0.000	0.000	0.610	0.260	0.100	0.030	0.000	0.072	1P	1/10	9	RW
2321	0.117	0.117	0.271	0.385	0.080	0.030	0.000	0.077	3P	1/10	9	54
2322	0.000	0.090	0.415	0.365	0.100	0.030	0.000	0.072	3P	1/10	9	27
2323	0.000	0.000	0.505	0.365	0.100	0.030	0.000	0.072	3P	1/10	9	RW
3321	0.054	0.054	0.182	0.580	0.100	0.030	0.000	0.077	SM	1/10	9	54
3322	0.000	0.840	0.206	0.530	0.150	0.030	0.000	0.072	SM	1/10	9	27
3323	0.000	0.000	0.290	0.530	0.150	0.030	0.000	0.072	SM	1/10	9	RW
4221	0.000	0.000	0.320	0.590	0.090	0.000	0.000	0.088	2M	1/6	9	54
4222	0.000	0.000	0.150	0.480	0.370	0.000	0.000	0.079	2M	1/6	9	27
4223	0.000	0.000	0.200	0.600	0.200	0.000	0.000	0.078	2M	1/6	9	RW
5122	0.000	0.000	0.000	0.770	0.200	0.030	0.000	0.079	3M	7/32	9	27
5123	0.000	0.000	0.000	0.770	0.200	0.030	0.000	0.075	3M	7/32	9	RW
1331	0.203	0.203	0.204	0.280	0.080	0.030	0.000	0.069	1P	1/10	10	54
1332	0.000	0.317	0.293	0.260	0.100	0.030	0.000	0.065	1P	1/10	10	27
1333	0.000	0.000	0.610	0.260	0.100	0.030	0.000	0.065	1P	1/10	10	RW
2331	0.117	0.117	0.271	0.385	0.080	0.030	0.000	0.069	3P	1/10	10	54
2332	0.000	0.090	0.415	0.365	0.100	0.030	0.000	0.065	3P	1/10	10	27
2333	0.000	0.000	0.505	0.365	0.100	0.030	0.000	0.065	3P	1/10	10	RW
3331	0.054	0.054	0.182	0.582	0.100	0.030	0.000	0.069	SM	1/10	10	54
3332	0.000	0.084	0.206	0.530	0.150	0.030	0.000	0.065	SM	1/10	10	27
3333	0.000	0.000	0.290	0.530	0.150	0.030	0.000	0.065	SM	1/10	10	RW
4331	0.000	0.030	0.290	0.590	0.090	0.000	0.000	0.069	2M	1/10	10	54
4332	0.000	0.000	0.150	0.480	0.370	0.000	0.000	0.065	2M	1/10	10	27
4333	0.000	0.000	0.200	0.600	0.200	0.000	0.000	0.065	2M	1/10	10	RW
4353	0.000	0.000	0.000	0.500	0.100	0.000	0.400	0.110	2M	1/10	4	RW
4231	0.000	0.000	0.320	0.590	0.090	0.000	0.000	0.079	2M	1/6	10	54
4232	0.000	0.000	0.150	0.480	0.370	0.000	0.000	0.071	2M	1/6	10	27
4233	0.000	0.000	0.200	0.600	0.200	0.000	0.000	0.071	2M	1/6	10	RW
5132	0.000	0.000	0.000	0.770	0.200	0.030	0.000	0.067	3M	7/32	10	27
5133	0.000	0.000	0.000	0.770	0.200	0.030	0.000	0.067	3M	7/32	10	RW

DRYDEV.DAT cont'd

1341	0.203	0.203	0.204	0.280	0.080	0.030	0.000	0.058	1P	1/10	12	54
1342	0.000	0.317	0.293	0.260	0.100	0.030	0.000	0.054	1P	1/10	12	27
1343	0.000	0.000	0.610	0.260	0.100	0.030	0.000	0.054	1P	1/10	12	RW
2341	0.117	0.117	0.271	0.385	0.080	0.030	0.000	0.058	3P	1/10	12	54
2342	0.000	0.090	0.415	0.365	0.100	0.030	0.000	0.054	3P	1/10	12	27
2343	0.000	0.000	0.505	0.365	0.100	0.030	0.000	0.054	3P	1/10	12	RW
3341	0.054	0.054	0.182	0.580	0.100	0.030	0.000	0.058	SM	1/10	12	54
3342	0.000	0.084	0.206	0.530	0.150	0.030	0.000	0.054	SM	1/10	12	27
3343	0.000	0.000	0.290	0.530	0.150	0.030	0.000	0.054	SM	1/10	12	RW
4341	0.000	0.030	0.290	0.590	0.090	0.000	0.000	0.058	2M	1/10	12	54
4342	0.000	0.000	0.150	0.480	0.370	0.000	0.000	0.054	2M	1/10	12	27
4343	0.000	0.000	0.200	0.600	0.200	0.000	0.000	0.054	2M	1/10	12	RW
4353	0.000	0.000	0.000	0.500	0.100	0.000	0.400	0.110	2M	1/10	4	RW
4242	0.000	0.000	0.300	0.575	0.125	0.000	0.000	0.059	2M	1/6	12	27
4243	0.000	0.000	0.200	0.600	0.200	0.000	0.000	0.059	2M	1/6	12	RW
5341	0.000	0.000	0.000	0.770	0.200	0.030	0.000	0.058	3M	1/10	12	54
5342	0.000	0.000	0.000	0.770	0.200	0.030	0.000	0.054	3M	1/10	12	27
5343	0.000	0.000	0.000	0.770	0.200	0.030	0.000	0.054	3M	1/10	12	RW
5353	0.000	0.000	0.000	0.700	0.200	0.000	0.100	0.110	3M	1/10	4	RW
5242	0.000	0.000	0.000	0.770	0.200	0.030	0.000	0.059	3M	1/6	12	27
5243	0.000	0.000	0.000	0.770	0.200	0.030	0.000	0.059	3M	1/6	12	RW
5253	0.000	0.000	0.000	0.700	0.200	0.000	0.100	0.111	3M	1/6	4	RW
5142	0.000	0.000	0.000	0.770	0.200	0.030	0.000	0.056	3M	7/32	12	27
5143	0.000	0.000	0.000	0.770	0.200	0.030	0.000	0.056	3M	7/32	12	RW

INVDEV.DAT Dry Inventory File

code	translation				code	translation			
5111	7/32	8	C	54	5441	1/12	12	C	54
5112	7/32	8	C	27	5442	1/12	12	C	27
5113	7/32	8	C	RW	5443	1/12	12	C	RW
5121	7/32	9	C	54	6111	7/32	8	D	54
5122	7/32	9	C	27	~~6112~~	~~7/32~~	~~8~~	~~D~~	~~27~~
5123	7/32	9	C	RW	6113	7/32	8	D	RW
5131	7/32	10	C	54	6121	7/32	9	D	54
5132	7/32	10	C	27	6122	7/32	9	D	27
5133	7/32	10	C	RW	6123	7/32	9	D	RW
5141	7/32	12	C	54	6131	7/32	10	D	54
5142	7/32	12	C	27	6132	7/32	10	D	27
5143	7/32	12	C	RW	6133	7/32	10	D	RW
5211	1/6	8	C	54	6141	7/32	12	D	54
5212	1/6	8	C	27	6142	7/32	12	D	27
5213	1/6	8	C	RW	6143	7/32	12	D	RW
5221	1/6	9	C	54	6211	1/6	8	D	54
5222	1/6	9	C	27	6212	1/6	8	D	27
5223	1/6	9	C	RW	6213	1/6	8	D	RW
5231	1/6	10	C	54	6221	1/6	9	D	54
5232	1/6	10	C	27	6222	1/6	9	D	27
5233	1/6	10	C	RW	6223	1/6	9	D	RW
5241	1/6	12	C	54	6231	1/6	10	D	54
5242	1/6	12	C	27	6232	1/6	10	D	27
5243	1/6	12	C	RW	6233	1/6	10	D	RW
5311	1/10	8	C	54	6241	1/6	12	D	54
5312	1/10	8	C	27	6242	1/6	12	D	27
5313	1/10	8	C	RW	6243	1/6	12	D	RW
5321	1/10	9	C	54	6311	1/10	8	D	54
5322	1/10	9	C	27	6312	1/10	8	D	27
5323	1/10	9	C	RW	6313	1/10	8	D	RW
5331	1/10	10	C	54	6321	1/10	9	D	54
5332	1/10	10	C	27	6322	1/10	9	D	27
5333	1/10	10	C	RW	6323	1/10	9	D	RW
5341	1/10	12	C	54	6331	1/10	10	D	54
5342	1/10	12	C	27	6332	1/10	10	D	27
5343	1/10	12	C	RW	6333	1/10	10	D	RW
5411	1/12	8	C	54	6341	1/10	12	D	54
5412	1/12	8	C	27	6342	1/10	12	D	27
5413	1/12	8	C	RW	6343	1/10	12	D	RW
5421	1/12	9	C	54	6411	1/12	8	D	54
5422	1/12	9	C	27	6412	1/12	8	D	27
5423	1/12	9	C	RW	6413	1/12	8	D	RW
5431	1/12	10	C	54	6421	1/12	9	D	54
5432	1/12	10	C	27	6422	1/12	9	D	27
5433	1/12	10	C	RW	6423	1/12	9	D	RW

INVDEV. DAT cont'd

6431	1/12	10	D	54
6432	1/12	10	D	27
6433	1/12	10	D	RW
6441	1/12	12	D	54
6442	1/12	12	D	27
6443	1/12	12	D	RW
7111	7/32	8	CP	54
7112	7/32	8	CP	27
7113	7/32	8	CP	RW
7121	7/32	9	CP	54
7122	7/32	9	CP	27
7123	7/32	9	CP	RW
7131	7/32	10	CP	54
7132	7/32	10	CP	27
7133	7/32	10	CP	RW
7141	7/32	12	CP	54
7142	7/32	12	CP	27
7143	7/32	12	CP	RW
7211	1/6	8	CP	54
7212	1/6	8	CP	27
7213	1/6	8	CP	RW
7221	1/6	9	CP	54
7222	1/6	9	CP	27
7223	1/6	9	CP	RW
7231	1/6	10	CP	54
7232	1/6	10	CP	27
7233	1/6	10	CP	RW
7241	1/6	12	CP	54
7242	1/6	12	CP	27
7243	1/6	12	CP	RW
7311	1/10	8	CP	54
7312	1/10	8	CP	27
7313	1/10	8	CP	RW
7321	1/10	9	CP	54
7322	1/10	9	CP	27
7323	1/10	9	CP	RW
7331	1/10	10	CP	54
7332	1/10	10	CP	27
7333	1/10	10	CP	RW
7341	1/10	12	CP	54
7342	1/10	12	CP	27
7343	1/10	12	CP	RW
7411	1/12	8	CP	54
7412	1/12	8	CP	27
7413	1/12	8	CP	RW
7421	1/12	9	CP	54
7422	1/12	9	CP	27
7423	1/12	9	CP	RW
7431	1/12	10	CP	54
7432	1/12	10	CP	27
7433	1/12	10	CP	RW
7441	1/12	12	CP	54
7442	1/12	12	CP	27
7443	1/12	12	CP	RW
9111	7/32	8	BP	54
9112	7/32	8	BP	27
9113	7/32	8	BP	RW
9121	7/32	9	BP	54
9122	7/32	9	BP	27
9123	7/32	9	BP	RW
9131	7/32	10	BP	54
9132	7/32	10	BP	27
9133	7/32	10	BP	RW
9141	7/32	12	BP	54
9142	7/32	12	BP	27
9143	7/32	12	BP	RW
9211	1/6	8	BP	54
9212	1/6	8	BP	27
9213	1/6	8	BP	RW
9221	1/6	9	BP	54
9222	1/6	9	BP	27
9223	1/6	9	BP	RW
9231	1/6	10	BP	54
9232	1/6	10	BP	27
9233	1/6	10	BP	RW
9241	1/6	12	BP	54
9242	1/6	12	BP	27
9243	1/6	12	BP	RW
9311	1/10	8	BP	54
9312	1/10	8	BP	27
9313	1/10	8	BP	RW
9321	1/10	9	BP	54
9322	1/10	9	BP	27
9323	1/10	9	BP	RW
9331	1/10	10	BP	54
9332	1/10	10	BP	27
9333	1/10	10	BP	RW
9341	1/10	12	BP	54
9342	1/10	12	BP	27
9343	1/10	12	BP	RW
9411	1/12	8	BP	54
9412	1/12	8	BP	27
9413	1/12	8	BP	RW
9421	1/12	9	BP	54
9422	1/12	9	BP	27
9423	1/12	9	BP	RW
9431	1/12	10	BP	54
9432	1/12	10	BP	27
9433	1/12	10	BP	RW
9441	1/12	12	BP	54
9442	1/12	12	BP	27
9443	1/12	12	BP	RW
10111	7/32	8	X	54
10112	7/32	8	X	27
10113	7/32	8	X	RW
10121	7/32	9	X	54
10122	7/32	9	X	27
10123	7/32	9	X	RW
10131	7/32	10	X	54
10132	7/32	10	X	27
10133	7/32	10	X	RW
10141	7/32	12	X	54
10142	7/32	12	X	27
10143	7/32	12	X	RW
10211	1/6	8	X	54
10212	1/6	8	X	27
10213	1/6	8	X	RW
10221	1/6	9	X	54
10222	1/6	9	X	27
10223	1/6	9	X	RW
10231	1/6	10	X	54
10232	1/6	10	X	27
10233	1/6	10	X	RW
10241	1/6	12	X	54
10242	1/6	12	X	27
10243	1/6	12	X	RW
10311	1/10	8	X	54
10312	1/10	8	X	27
10313	1/10	8	X	RW
10321	1/10	9	X	54
10322	1/10	9	X	27
10323	1/10	9	X	RW
10331	1/10	10	X	54
10332	1/10	10	X	27
10333	1/10	10	X	RW
10341	1/10	12	X	54
10342	1/10	12	X	27
10343	1/10	12	X	RW
10411	1/12	8	X	54
10412	1/12	8	X	27
10413	1/12	8	X	RW
10421	1/12	9	X	54
10422	1/12	9	X	27
10423	1/12	9	X	RW
10431	1/12	10	X	54
10432	1/12	10	X	27
10433	1/12	10	X	RW
10441	1/12	12	X	54
10442	1/12	12	X	27
10443	1/12	12	X	RW
2111	7/32	8	DCLR	54
2112	7/32	8	DCLR	27
2121	7/32	9	DCLR	54
2122	7/32	9	DCLR	27
2131	7/32	10	DCLR	54
2132	7/32	10	DCLR	27
2141	7/32	12	DCLR	54
~~2142~~	7/32	12	DCLR	27
2211	1/6	8	DCLR	54
2212	1/6	8	DCLR	27
2221	1/6	9	DCLR	54
2222	1/6	9	DCLR	27
2231	1/6	10	DCLR	54
2232	1/6	10	DCLR	27
2241	1/6	12	DCLR	54
2242	1/6	12	DCLR	27
2311	1/10	8	DCLR	54
2312	1/10	8	DCLR	27
2321	1/10	9	DCLR	54
2322	1/10	9	DCLR	27
2331	1/10	10	DCLR	54
2332	1/10	10	DCLR	27
2341	1/10	12	DCLR	54
2342	1/10	12	DCLR	27
2411	1/12	8	DCLR	54
2412	1/12	8	DCLR	27

INVDEV. DAT cont'd

2421	1/12	9	DCLR	54
2422	1/12	9	DCLR	27
2431	1/12	10	DCLR	54
2432	1/12	10	DCLR	27
2441	1/12	12	DCLR	54
2442	1/12	12	DCLR	27
1111	7/32	8	ECLR	54
1121	7/32	9	ECLR	54
1131	7/32	10	ECLR	54
1141	7/32	12	ECLR	54
1211	1/6	8	ECLR	54
1221	1/6	9	ECLR	54
1231	1/6	10	ECLR	54
1241	1/6	12	ECLR	54
1311	1/10	8	ECLR	54
1321	1/10	9	ECLR	54
1331	1/10	10	ECLR	54
1341	1/10	12	ECLR	54
1411	1/12	8	ECLR	54
1421	1/12	9	ECLR	54
1431	1/12	10	ECLR	54
1441	1/12	12	ECLR	54
3111	7/32	8	A	54
3121	7/32	9	A	54
3131	7/32	10	A	54
3141	7/32	12	A	54
3211	1/6	8	A	54
3221	1/6	9	A	54
3231	1/6	10	A	54
3241	1/6	12	A	54
3311	1/10	8	A	54
3321	1/10	9	A	54
3331	1/10	10	A	54
3341	1/10	12	A	54
3411	1/12	8	A	54
3421	1/12	9	A	54
3431	1/12	10	A	54
3441	1/12	12	A	54
4111	7/32	8	B	54
4121	7/32	9	B	54
4131	7/32	10	B	54
4141	7/32	12	B	54
4211	1/6	8	B	54
4221	1/6	9	B	54
4231	1/6	10	B	54
4241	1/6	12	B	54
4311	1/10	8	B	54
4321	1/10	9	B	54
4331	1/10	10	B	54
4341	1/10	12	B	54
4411	1/12	8	B	54
4421	1/12	9	B	54
4431	1/12	10	B	54
4441	1/12	12	B	54
8111	7/32	8	M	54
8121	7/32	9	M	54
8131	7/32	10	M	54
8141	7/32	12	M	54
8211	1/6	8	M	54
8221	1/6	9	M	54
8231	1/6	10	M	54
8241	1/6	12	M	54
8311	1/10	8	M	54
8321	1/10	9	M	54
8331	1/10	10	M	54
8341	1/10	12	M	54
8411	1/12	8	M	54
8421	1/12	9	M	54
8431	1/12	10	M	54
8441	1/12	12	M	54
5151	7/32	4	C	54
5152	7/32	4	C	27
5153	7/32	4	C	RW
5251	1/6	4	C	54
5252	1/6	4	C	27
5253	1/6	4	C	RW
5351	1/10	4	C	54
5352	1/10	4	C	27
5353	1/10	4	C	RW
5451	1/12	4	C	54
5452	1/12	4	C	27
5453	1/12	4	C	RW
6151	7/32	4	D	54
6152	7/32	4	D	27
6153	7/32	4	D	RW
6251	1/6	4	D	54
6252	1/6	4	D	27
6253	1/6	4	D	RW
6351	1/10	4	D	54
6352	1/10	4	D	27
6353	1/10	4	D	RW
6451	1/12	4	D	54
6452	1/12	4	D	27
6453	1/12	4	D	RW
11153	7/32	4	SOL	RW
11253	1/6	4	SOL	RW
11353	1/10	4	SOL	RW
11453	1/12	4	SOL	RW

APPENDIX C

RESULTS

APPENDIX C.1 TRIAL ONE RESULTS

TRIAL RUN #1

PROPOSED MILL SCHEDULE
FOR SPREADER OPERATION

MACHINE	PANEL SIZE	PANEL TYPE	PANEL GRADE	QUANTITY
1	4 X 8	SIDING	5/8 Premium Solid Center	9
1	4 X 8	SIDING	5/8 Sound C Center	50
1	4 X 8	SIDING	3/8 Regular	3
1	4 X 8	SIDING	3/8 Premium	3
1	4 X 8	SIDING	3/8 Premium	25
1	4 X 8	SIDING	3/8 Sound	25
1	4 X 8	SIDING	5/8 Premuim C Center	60
1	4 X 8	SIDING	5/8 Sound C Center	50
1	4 X 8	SIDING	3/8 Premium	20
1	4 X 8	SIDING	3/8 Sound	25
1	4 X 8	SIDING	5/8 Sound Solid Center	40
1	4 X 8	SIDING	5/8 Premium Solid Center	25
1	4 X 8	SIDING	5/8 Sound C Center	25
2	4 X 9	SIDING	5/8 Premium C Center	50
2	4 X 9	SIDING	3/8 Sound	10
2	4 X 9	SIDING	5/8 Sound C Center	25
2	4 X 9	SIDING	5/8 Sound Solid Center	25
2	4 X 9	SIDING	5/8 Premium Solid Center	6
2	4 X 9	SIDING	5/8 Premium C Center	30
2	4 X 9	SIDING	5/8 Premium Solid Center	26
2	4 X 9	SIDING	3/8 Sound	10
2	4 X 9	SIDING	5/8 Sound C Center	40
2	4 X 9	SIDING	5/8 Premium C Center	12
2	4 X 9	SIDING	5/8 Premium Solid Center	25
2	4 X 9	SIDING	5/8 Premium C Center	30
2	4 X 9	SIDING	3/8 Sound	10
2	4 X 9	SIDING	5/8 Sound C Center	45
2	4 X 9	SIDING	5/8 Sound Solid Center	40
2	4 X 9	SIDING	5/8 Premium Solid Center	3
2	4 X 9	SIDING	5/8 Premium C Center	60
3	4 X 8	SHEATHING	3/4 CD	64
3	4 X 8	SANDED	13/16 AC	14
3	4 X 9	SANDED	13/16 AC	35
3	4 X 8	SANDED	13/16 AB	7
3	4 X 8	SANDED	13/16 AC	2
3	4 X 8	SHEATHING	5/8 CD	60
3	4 X 8	SANDED	13/16 AB	25
3	4 X 8	SANDED	7/16 AB	5
3	4 X 8	SHEATHING	3/8 CC Plug	25
3	4 X 8	SHEATHING	5/8 CD	60
3	4 X 8	SANDED	5/16 AA	3.5
3	4 X 8	SANDED	13/16 AA	7
3	4 X 8	SHEATHING	3/8 CC Plug	40
4	4 X 8	DURAPLY	1/2 G2S	36
4	4 X 8	DURAPLY	5/8 G1S	55
4	4 X 8	DURAPLY	5/8 G1S	17
4	4 X 8	MARINE	9/16 AA	14
4	4 X 8	MARINE	5/16 AA	3.5
4	4 X 8	MARINE	13/16 AA	12
4	4 X 8	DURAPLY	5/8 G1S	55
4	4 X 8	DURAPLY	3/8 G1S	40
4	4 X 8	DURAPLY	5/16 G2S	11
6	4 X 10	SHEATHING	13/16 CC Plug	25
6	4 X 10	SANDED	13/16 AC	35
6	4 X 10	SIDING	5/8 Premium C Center	30
6	4 X 10	SIDING	5/8 Sound Solid Center	20
6	4 X 10	SANDED	5/16 AC	40
6	4 X 12	SANDED	13/16 AC	1
6	4 X 12	SHEATHING	5/16 CC Plug	33

TRIAL RUN #1

GREEN VENEER CODE SEQUENCE

NO.	CODE	NO.	CODE	NO.	CODE
1	14313	56	15132	111	14211
2	12311	57	12331	112	15213
3	12311	58	22213	113	15213
4	12311	59	12311	114	13353
5	15113	60	12311	115	13353
6	14242	61	14213	116	22212
7	15341	62	15113	117	13353
8	14341	63	14221	118	22212
9	14243	64	12311	119	23232
10	22212	65	15113	120	23232
11	15343	66	15213	121	15112
12	14341	67	14221	122	15213
13	14341	68	12321		
14	14341	69	15213		
15	14341	70	15213		
16	12413	71	15213		
17	22212	72	14221		
18	12311	73	14211		
19	14331	74	15213		
20	14331	75	22222		
21	12311	76	12321		
22	12311	77	15211		
23	12311	78	15213		
24	12311	79	22222		
25	12413	80	12321		
26	12311	81	15211		
27	12311	82	15113		
28	13333	83	15211		
29	13331	84	15113		
30	13331	85	14211		
31	14221	86	12311		
32	13321	87	15213		
33	13321	88	22222		
34	14213	89	15213		
35	22212	90	22222		
36	12311	91	15113		
37	12311	92	14211		
38	15213	93	12311		
39	23232	94	15113		
40	12311	95	12311		
41	15213	96	15113		
42	23232	97	14211		
43	15213	98	15213		
44	14221	99	14211		
45	12321	100	15213		
46	15213	101	14211		
47	15143	102	15213		
48	14231	103	22222		
49	14231	104	15213		
50	12331	105	14211		
51	14243	106	15213		
52	15132	107	15213		
53	15213	108	15213		
54	12311	109	15213		
55	14213	110	15213		

TRIAL RUN #1

VENEER INVENTORY RESULTS

(IN UNIT LOADS)

VENEER GRADE				MINIMUM	MAXIMUM
7/32	8	C	27	0.00	7.92
7/32	8	C	RW	0.00	14.01
7/32	10	C	27	0.00	4.13
7/32	12	C	RW	0.00	11.02
1/6	8	C	54	0.00	9.65
1/6	8	C	27	0.00	36.56
1/6	8	C	RW	0.00	46.20
1/6	9	C	54	0.00	12.67
1/6	9	C	27	0.00	51.45
1/6	10	C	54	0.00	5.23
1/6	10	C	27	0.00	11.10
1/6	12	C	27	0.00	1.73
1/6	12	C	RW	0.00	8.13
1/10	8	C	54	0.00	114.06
1/10	8	C	RW	0.00	6.41
1/10	9	C	54	0.00	50.38
1/10	10	C	54	0.00	52.10
1/10	10	C	RW	0.00	4.19
1/10	12	C	54	0.00	72.90
1/10	12	C	RW	-0.01	5.07
1/12	8	C	RW	0.00	2.22
7/32	8	D	54	0.00	7.49
7/32	8	D	27	0.00	2.06
7/32	8	D	RW	-0.03	10.14
7/32	10	D	27	0.00	1.67
7/32	12	D	RW	0.00	2.86
1/6	8	D	54	0.00	8.36
1/6	8	D	27	0.00	3.18
1/6	8	D	RW	-0.06	41.59
1/6	9	D	54	-0.02	2.44
1/6	9	D	27	0.00	4.47
1/6	10	D	54	0.00	0.80
1/6	10	D	27	0.00	1.14
1/6	12	D	27	0.00	0.37
1/6	12	D	RW	0.00	3.31
1/10	8	D	54	0.00	27.49
1/10	8	D	RW	0.00	2.25
1/10	9	D	54	0.00	13.24
1/10	10	D	54	0.00	9.52
1/10	10	D	RW	0.00	1.19
1/10	12	D	54	0.00	11.35
1/10	12	D	RW	0.00	1.32
1/12	8	D	RW	0.00	0.61
1/6	8	CP	54	0.00	6.08
1/6	8	CP	27	-0.01	7.17

TRIAL RUN #1

VENEER INVENTORY RESULTS

(IN UNIT LOADS)

VENEER GRADE				MINIMUM	MAXIMUM
1/6	9	CP	54	0.00	7.47
1/6	9	CP	27	-0.04	7.44
1/6	10	CP	54	0.00	1.81
1/6	10	CP	27	0.00	2.41
1/10	8	CP	54	-0.06	2.21
1/10	10	CP	54	-0.03	0.65
1/10	12	CP	54	-0.01	1.46
1/6	8	BP	54	0.00	25.15
1/6	8	BP	RW	0.00	15.68
1/6	9	BP	54	0.00	16.65
1/6	10	BP	54	0.00	2.84
1/6	12	BP	27	0.00	0.90
1/6	12	BP	RW	0.00	3.31
1/10	8	BP	54	0.00	38.17
1/10	8	BP	RW	0.00	2.59
1/10	9	BP	54	0.00	25.01
1/10	10	BP	54	0.00	11.98
1/10	10	BP	RW	0.00	2.29
1/10	12	BP	54	0.00	30.08
1/12	8	BP	RW	0.00	3.07
7/32	8	X	27	0.00	0.31
7/32	8	X	RW	0.00	0.55
7/32	10	X	27	0.00	0.25
7/32	12	X	RW	0.00	0.43
1/6	8	X	54	-0.03	3.13
1/6	8	X	RW	0.00	1.89
1/6	9	X	54	0.00	0.65
1/10	8	X	54	0.00	8.85
1/10	9	X	54	0.00	4.41
1/10	10	X	54	0.00	2.97
1/10	10	X	RW	0.00	0.24
1/10	12	X	54	0.00	0.09
1/10	12	X	RW	0.00	0.20
1/12	8	X	RW	0.00	0.18
1/10	8	DCLR	54	0.00	34.72
1/10	9	DCLR	54	0.00	4.02
1/10	10	DCLR	54	0.00	8.43
1/10	12	DCLR	54	0.00	3.58
1/10	8	ECLR	54	0.00	39.57
1/10	9	ECLR	54	0.00	4.72
1/10	10	ECLR	54	0.00	8.25
1/10	8	A	54	0.00	30.37
1/10	9	A	54	0.00	5.11
1/10	10	A	54	0.00	6.49
1/10	12	A	54	0.00	23.45

TRIAL RUN #1

VENEER INVENTORY RESULTS

(IN UNIT LOADS)

VENEER GRADE	MINIMUM	MAXIMUM
1/10 8 B 54	0.00	5.29
1/10 9 B 54	-0.01	0.27
1/10 10 B 54	-0.01	1.21
1/10 12 B 54	0.00	4.41
1/10 8 M 54	-0.01	2.47
1/10 4 C RW	0.00	7.76
1/10 4 D RW	0.00	5.17
1/10 4 SOL RW	0.00	6.44

CBP-LEBANON

TASK NO. ________

SPREADER ORDER FILE

11-JUN-84

TASK NO.	RATE	PANEL GRADE					CONSTRUCTION METHOD										PRODUCTION QUANTITY
121141	2.08	8	SID	PREM-SCR	EXT	5/8	1	1	1	0	2	231	531	751	51	55	816
151141	2.08	8	SID	SND-CCTR	EXT	5/8	1	1	1	0	2	331	531	551	51	55	3000
111131	4.17	8	SID	REGULAR	EXT	3/8	1	1	0	0	1	131	531	51	51	52	144
161121	4.17	8	SID	PREMIUM	EXT	3/8	1	1	0	0	1	231	531	51	51	52	2304
171121	4.17	8	SID	SND	EXT	3/8	1	1	0	0	1	331	531	51	51	52	2400
131141	2.08	8	SID	PREM-CCR	EXT	5/8	1	1	1	0	2	231	531	551	51	55	1440
141141	2.08	8	SID	SND-SCTR	EXT	5/8	1	1	1	0	2	331	531	751	51	55	960
512151	2.08	8	SHE	CDX	INT	3/4	1	1	1	0	2	531	631	611	51	62	1280
231151	2.08	8	SAN	AC	EXT	3/4	1	1	1	0	2	331	531	521	51	51	320
221151	2.08	8	SAN	AB	EXT	3/4	1	1	1	0	2	331	431	521	51	51	640
512141	1.87	8	SHE	CDX	INT	5/8	1	1	0	0	2	521	621	51	51	65	2400
221121	4.17	8	SAN	AB	EXT	3/8	1	1	0	0	1	331	431	51	51	51	200
541122	4.17	8	SHE	CC-PTS	EXT	3/8	1	1	0	0	1	731	531	51	51	52	2600
211111	4.17	8	SAN	AA	EXT	1/4	1	1	0	0	1	331	331	51	51	53	140
211151	2.08	8	SAN	AA	EXT	3/4	1	1	1	0	2	331	331	521	51	51	140
421131	1.87	8	DUR	G2S	EXT	1/2	1	1	1	0	2	431	431	531	51	53	720
411141	1.87	8	DUR	G1S	EXT	5/8	1	1	1	0	2	431	531	531	51	52	2540
311131	2.08	8	MAR	AA	EXT	1/2	1	1	1	0	2	831	831	721	51	73	280
311111	4.17	8	MAR	AA	EXT	1/4	1	1	0	0	1	831	831	51	51	73	70
311151	1.46	8	MAR	AA	EXT	3/4	1	1	2	0	3	831	831	721	51	73	240
411121	3.75	8	DUR	G1S	EXT	3/8	1	1	0	0	1	431	531	51	51	52	1600
421111	1.87	8	DUR	G2S	EXT	1/4	1	1	0	0	1	431	431	51	51	54	220
131241	2.08	9	SID	PREM-CCR	EXT	5/8	1	1	1	0	2	233	532	522	51	56	3640
171221	4.17	9	SID	SND	EXT	3/8	1	1	0	0	1	332	532	51	51	52	1200
151241	2.08	9	SID	SND-CCTR	EXT	5/8	1	1	1	0	2	332	532	522	51	56	2300
141241	2.08	9	SID	SND-SCTR	EXT	5/8	1	1	1	0	2	332	532	722	51	56	1300
121241	2.08	9	SID	PREM-SCR	EXT	5/8	1	1	1	0	2	232	532	722	51	56	1200
231251	2.08	9	SAN	AC	EXT	3/4	1	1	1	0	2	332	532	522	51	51	700
541352	1.75	10	SHE	CC-PTS	EXT	3/4	1	1	1	0	2	733	533	513	51	52	500
231351	1.75	10	SAN	AC	EXT	3/4	1	1	1	0	2	333	533	523	51	51	700
131341	1.75	10	SID	PREM-CCR	EXT	5/8	1	1	1	0	2	233	533	523	51	56	600
141341	1.75	10	SID	SND-SCTR	EXT	5/8	1	1	1	0	2	333	533	723	51	56	400
231311	3.50	10	SAN	AC	EXT	1/4	1	1	0	0	1	333	533	51	51	53	1600
231451	1.46	12	SAN	AC	EXT	3/4	1	1	1	0	2	334	534	524	51	51	30
541401	2.93	12	SHE	CC-PTS	EXT	5/16	1	1	0	0	1	734	534	51	51	53	1320

CBP-LEBANON CURRENT INVENTORY 11-JUN-84

DRY INVENTORY:

GRADE	THK	LENGTH	WIDTH	QUANTITY

240

DRY-END SCHEDULE
CBP-LEBANON

11-JUN-84

SPREADER
OPERATION
MACHINE:

DAY / SHIFT	1 QTY	THK	LEN	GRADE	CONST	2 QTY	THK	LEN	GRADE	CONST	3 QTY	THK	LEN	GRADE	CONST	4 QTY	THK	LEN	GRADE	CONST
DAY: 4 SHIFT: GYD	816	5/8	8	SID PREM-SCR	EXT 1						30	3/4	12	SAN AC	EXT 1	720	1/2	8	DUR G2S	EXT 1
	182	5/8	8	SID SND-SCTR	EXT 1						1320	5/16	12	SHE CC-PTS	EXT 1	177	1/4	8	DUR G2S	EXT 1
											25	1/4	10	SAN AC	EXT 1					
DAY: 4 SHIFT: DAYS	778	5/8	8	SID SND-SCTR	EXT 1	998	5/8	9	SID PREM-CCR	EXT 1	1575	1/4	10	SAN AC	EXT 1	43	1/4	8	DUR G2S	EXT 1
	442	3/8	8	SID SND	EXT 1						52	5/8	10	SID SND-SCTR	EXT 1	1600	3/8	8	DUR G1S	EXT 1
																57	5/8	8	DUR G1S	EXT 1
DAY: 4 SHIFT: SWING	1958	3/8	8	SID SND	EXT 1	998	5/8	9	SID PREM-CCR	EXT 1	68	5/8	10	SID SND-SCTR	EXT 1	897	5/8	8	DUR G1S	EXT 1
	44	3/8	8	SHE CC-PTS	EXT 2						700	3/4	10	SAN AC	EXT 1					
											72	3/4	10	SHE CC-PTS	EXT 2					
DAY: 5 SHIFT: GYD	889	3/8	8	SHE CC-PTS	EXT 2						428	3/4	10	SHE CC-PTS	EXT 2	897	5/8	8	DUR G1S	EXT 1
	1111	3/8	8	SID PREMIUM	EXT 1						490	3/4	9	SAN AC	EXT 1					
DAY: 5 SHIFT: DAYS	1193	3/8	8	SID PREMIUM	EXT 1	998	5/8	9	SID PREM-CCR	EXT 1	210	3/4	9	SAN AC	EXT 1	689	5/8	8	DUR G1S	EXT 1
	144	3/8	8	SID REGULAR	EXT 1															
	298	5/8	8	SHE CDX	INT 1															
DAY: 5 SHIFT: SWING	897	5/8	8	SHE CDX	INT 1	646	5/8	9	SID PREM-CCR	EXT 1										
						353	5/8	9	SID PREM-SCR	EXT 1										
DAY: 6 SHIFT: GYD	897	5/8	8	SHE CDX	INT 1															

241

DAY: 6 SHIFT: DAYS			
308 5/8 8 SHE CDX INT 1	847 5/8 9 SID PREM-SCR EXT 1		
140 1/4 8 SAN AA EXT 1	151 5/8 9 SID SND-SCTR EXT 1		
140 3/4 8 SAN AA EXT 1			
447 3/4 8 SAN AB EXT 1			
DAY: 6 SHIFT: SWING			
193 3/4 8 SAN AB EXT 1	998 5/8 9 SID SND-SCTR EXT 1		
200 3/8 8 SAN AB EXT 1			
320 3/4 8 SAN AC EXT 1			
386 5/8 8 SID SND-CCTR EXT 1			
DAY: 7 SHIFT: GYD			
998 5/8 8 SID SND-CCTR EXT 1			
DAY: 7 SHIFT: DAYS			
998 5/8 8 SID SND-CCTR EXT 1	151 5/8 9 SID SND-SCTR EXT 1		
	848 5/8 9 SID SND-CCTR EXT 1		
DAY: 7 SHIFT: SWING			
618 5/8 8 SID SND-CCTR EXT 1	998 5/8 9 SID SND-CCTR EXT 1		
381 5/8 8 SID PREM-CCR EXT 1			
DAY: 8 SHIFT: GYD			
998 5/8 8 SID PREM-CCR EXT 1			
DAY: 8 SHIFT: DAYS			
61 5/8 8 SID PREM-CCR EXT 1	354 5/8 9 SID SND-CCTR EXT 1		
70 1/4 8 MAR AA EXT 1	1200 3/8 9 SID SND EXT 1		
240 3/4 8 MAR AA EXT 1	39 5/8 10 SID SND-SCTR EXT 1		
280 1/2 8 MAR AA EXT 1			
281 3/4 8 SHE CDX INT 1			
DAY: 8 SHIFT: SWING			
998 3/4 8 SHE CDX INT 1	240 5/8 10 SID SND-SCTR EXT 1		
	599 5/8 10 SID PREM-CCR EXT 1		

MACHINE:

DAY / SHIFT	5: QTY	THK	LEN	GRADE	CONST	6: QTY	THK	LEN	GRADE	CONST
DAY: 4 SHIFT: GYD										
DAY: 4 SHIFT: DAYS						1666	3/8	8	SHE	CC-PTS EXT 2
DAY: 4 SHIFT: SWING										
DAY: 5 SHIFT: GYD										
DAY: 5 SHIFT: DAYS										
DAY: 5 SHIFT: SWING										
DAY: 6 SHIFT: GYD										
DAY: 6 SHIFT: DAYS										

DAY: 6 SHIFT: SWING		
DAY: 7 SHIFT: GYD		
DAY: 7 SHIFT: DAYS		
DAY: 7 SHIFT: SWING		
DAY: 8 SHIFT: GYD		
DAY: 8 SHIFT: DAYS		
DAY: 8 SHIFT: SWING		

PATCHER
OPERATION
MACHINE:

Machine 1

DAY / SHIFT	QTY	THK	LEN	WID	GRADE	COMPL
DAY: 1 SHIFT: DAYS						
DAY: 1 SHIFT: SWING						
DAY: 2 SHIFT: GYD						
DAY: 2 SHIFT: DAYS	1.40	1/10	8	54	BP	3 DAYS
DAY: 2 SHIFT: SWING	2.24	1/10	8	54	BP	3 DAYS
DAY: 3 SHIFT: GYD						
DAY: 3 SHIFT: DAYS	0.56	1/6	9	54	D	4 GYD
DAY: 3 SHIFT: SWING	0.18	1/6	9	54	D	4 GYD
	1.09	1/6	9	54	D	4 GYD
DAY: 4 SHIFT: GYD	1.01	1/10	9	54	BP	4 SWG

Machine 2

DAY / SHIFT	QTY	THK	LEN	WID	GRADE	COMPL
DAY: 1 SHIFT: DAYS						
DAY: 1 SHIFT: SWING						
DAY: 2 SHIFT: GYD						
DAY: 2 SHIFT: DAYS	1.93	1/10	8	54	BP	3 DAYS
DAY: 2 SHIFT: SWING	1.72	1/10	8	54	BP	3 DAYS
	0.74	1/10	8	54	BP	4 DAYS
DAY: 3 SHIFT: GYD	2.46	1/10	8	54	BP	4 DAYS
DAY: 3 SHIFT: DAYS	2.46	1/10	8	54	BP	4 DAYS
DAY: 3 SHIFT: SWING	2.34	1/10	8	54	BP	4 DAYS
DAY: 4 SHIFT: GYD						

Machine 3

DAY / SHIFT	QTY	THK	LEN	WID	GRADE	COMPL
DAY: 1 SHIFT: DAYS						
DAY: 1 SHIFT: SWING						
DAY: 2 SHIFT: GYD						
DAY: 2 SHIFT: DAYS						
DAY: 2 SHIFT: SWING	0.15	1/10	12	54	BP	3 GYD
	0.22	1/10	8	54	BP	3 GYD
	2.72	1/6	8	27	CP	3 SWG
DAY: 3 SHIFT: GYD	3.07	1/6	8	27	CP	3 SWG
DAY: 3 SHIFT: DAYS						
DAY: 3 SHIFT: SWING	1.24	1/10	8	54	BP	4 SWG
DAY: 4 SHIFT: GYD	2.24	1/10	8	54	BP	4 SWG

Machine 4

DAY / SHIFT	QTY	THK	LEN	WID	GRADE	COMPL
DAY: 1 SHIFT: DAYS						
DAY: 1 SHIFT: SWING						
DAY: 2 SHIFT: GYD						
DAY: 2 SHIFT: DAYS	3.01	1/6	8	27	CP	3 DAYS
DAY: 2 SHIFT: SWING	3.07	1/6	8	27	CP	3 DAYS
DAY: 3 SHIFT: GYD						
DAY: 3 SHIFT: DAYS						
DAY: 3 SHIFT: SWING	1.47	1/10	9	54	BP	4 DAYS
DAY: 4 SHIFT: GYD	0.89	1/10	9	54	BP	4 DAYS

DAY: 4 SHIFT: DAYS			
DAY: 4 SHIFT: SWING			
DAY: 5 SHIFT: GYD			
1.13 1/10 8 54 BP 5 SWG			
DAY: 5 SHIFT: DAYS			
1.14 1/10 8 54 BP 5 SWG		0.76 1/10 8 54 BP 6 GYD	0.37 1/6 8 RW D 5 SWG 1.13 1/6 9 27 CP 6 GYD 0.39 1/6 8 54 D 6 GYD
DAY: 5 SHIFT: SWING			
		0.26 1/10 8 54 BP 6 GYD 0.24 1/6 8 54 D 6 SWG	0.02 1/6 8 54 D 6 GYD 0.41 1/6 8 54 D 6 GYD 1.83 1/10 9 54 BP 6 SWG
DAY: 6 SHIFT: GYD			
1.13 1/6 9 27 CP 6 SWG 0.83 1/6 8 RW D 7 GYD		2.44 1/6 8 54 D 6 SWG	2.12 1/10 9 54 BP 6 SWG
DAY: 6 SHIFT: DAYS			
2.97 1/6 8 RW D 7 GYD			0.13 1/10 9 54 BP 6 SWG
DAY: 6 SHIFT: SWING			
DAY: 7 SHIFT: GYD			
	1.63 1/10 8 54 ECLR 7 SWG	0.91 1/6 8 27 CP 7 SWG	
DAY: 7 SHIFT: DAYS			
	0.86 1/10 8 54 ECLR 7 SWG	1.18 1/6 8 27 CP 7 SWG	
DAY: 7 SHIFT: SWING			

MACHINE:

DAY	SHIFT	5 (QTY THK LEN WID GRADE COMPL)	6 (QTY THK LEN WID GRADE COMPL)	7 (QTY THK LEN WID GRADE COMPL)	8 (QTY THK LEN WID GRADE COMPL)
1	DAYS	3.36 1/10 12 54 BP 3 DAYS	3.36 1/10 12 54 BP 3 DAYS	3.36 1/10 12 54 BP 3 DAYS	
1	SWING	1.68 1/10 12 54 BP 3 DAYS	1.68 1/10 12 54 BP 3 DAYS	1.68 1/10 12 54 BP 3 DAYS	
2	GYD	1.68 1/10 12 54 BP 3 DAYS	1.68 1/10 12 54 BP 3 DAYS	1.68 1/10 12 54 BP 3 DAYS	0.26 1/10 12 54 BP 3 GYD
2	DAYS	1.68 1/10 12 54 BP 3 DAYS	1.68 1/10 12 54 BP 3 DAYS	1.68 1/10 12 54 BP 3 DAYS	1.41 1/10 12 54 BP 3 GYD 0.39 1/10 8 54 BP 4 DAYS
2	SWING	1.60 1/10 13 54 BP 3 DAYS	1.60 1/10 12 54 BP 3 DAYS	1.60 1/10 12 54 BP 3 DAYS	2.46 1/10 8 54 BP 4 DAYS
3	GYD				2.46 1/10 8 54 BP 4 DAYS
3	DAYS				2.46 1/10 8 54 BP 4 DAYS
3	SWING				2.24 1/10 8 54 BP 4 DAYS
4	GYD				
4	DAYS				

DAY: 4 SHIFT: SWING			
DAY: 5 SHIFT: GYD			
DAY: 5 SHIFT: DAYS			
DAY: 5 SHIFT: SWING			
DAY: 6 SHIFT: GYD			
DAY: 6 SHIFT: DAYS			
DAY: 6 SHIFT: SWING			
DAY: 7 SHIFT: GYD			
DAY: 7 SHIFT: DAYS			
DAY: 7 SHIFT: SWING			

MACHINE:

DAY / SHIFT	9 QTY	9 THK	9 LEN	9 WID	9 GRADE	9 COMPL	10 QTY	10 THK	10 LEN	10 WID	10 GRADE	10 COMPL	11 QTY	11 THK	11 LEN	11 WID	11 GRADE	11 COMPL	12 QTY	12 THK	12 LEN	12 WID	12 GRADE	12 COMPL
DAY: 1 SHIFT: DAYS																								
DAY: 1 SHIFT: SWING																								
DAY: 2 SHIFT: GYD																								
DAY: 2 SHIET: DAYS							0.90	1/10	8	54	BP	2 SWG												
							0.26	1/10	10	54	BP	3 GYD												
							0.41	1/6	9	54	D	3 GYD												
							0.48	1/10	10	54	BP	4 DAYS												
DAY: 2 SHIET: SWING	0.74	1/10	8	54	BP	3 DAYS	2.12	1/10	10	54	BP	4 DAYS	0.87	1/6	8	27	CP	3 DAYS						
DAY: 3 SHIET: GYD	0.16	1/10	8	54	BP	3 DAYS	2.12	1/10	10	54	BP	4 DAYS	0.48	1/6	8	27	CP	3 DAYS	0.48	1/10	8	54	BP	3 DAYS
	2.30	1/10	8	54	BP	4 GYD							0.50	1/6	10	27	CP	3 DAYS	0.39	1/10	8	54	BP	3 SWG
													1.44	1/10	10	54	BP	4 GYD	0.40	1/6	10	27	CP	3 SWG
																			0.21	1/10	10	54	BP	3 SWG
DAY: 3 SHIET: DAYS	2.24	1/10	8	54	BP	4 GYD	2.12	1/10	10	54	BP	4 DAYS	2.10	1/10	10	54	BP	4 GYD	0.14	1/10	10	54	BP	3 SWG
																			2.30	1/10	8	54	BP	4 DAYS
DAY: 3 SHIET: SWING							1.93	1/10	10	54	BP	4 DAYS							2.24	1/10	8	54	BP	4 DAYS
DAY: 4 SHIET: GYD	1.23	1/6	8	RW	D	4 SWG																		
DAY: 4 SHIET: DAYS	1.88	1/6	8	RW	D	4 SWG							0.75	1/6	9	27	CP	5 GYD						

DAY: 4 SHIFT: SWING			
0.75 1/10 9 54 BP 6 GYD		1.88 1/6 9 27 CP 5 GYD 1.01 1/6 9 27 CP 6 GYD	3.24 1/6 9 27 CP 5 SWG
DAY: 5 SHIFT: GYD			
2.12 1/10 9 54 BP 6 GYD		3.36 1/6 9 27 CP 6 GYD	3.07 1/6 9 27 CP 5 SWG
DAY: 5 SHIFT: DAYS			
1.93 1/10 9 54 BP 6 GYD		3.07 1/6 9 27 CP 6 GYD	
DAY: 5 SHIFT: SWING			
DAY: 6 SHIFT: GYD			
DAY: 6 SHIFT: DAYS			
DAY: 6 SHIFT: SWING			
DAY: 7 SHIFT: GYD			
	1.62 1/10 8 54 ECLR 7 SWG		2.12 1/10 8 54 ECLR 7 SWG
DAY: 7 SHIFT: DAYS			
	0.86 1/10 8 54 ECLR 7 SWG		0.02 1/10 8 54 ECLR 7 SWG
DAY: 7 SHIFT: SWING			

MACHINE:

	13	14	15	16
	QTY THK LEN WID GRADE COMPL	QTY THK LEN WID GRADE COMPL	QTY THK LEN WID GRADE COMPL	QTY THK LEN WID GRADE COMPL
DAY: 1 SHIFT: DAYS				
DAY: 1 SHIFT: SWING			0.39 1/10 8 54 BP 3 SWG	0.39 1/10 8 54 BP 3 SWG
DAY: 2 SHIFT: GYD	0.45 1/10 8 54 BP 2 SWG 0.83 1/10 8 54 BP 3 SWG	0.13 1/10 10 54 BP 2 DAYS 0.23 1/10 8 54 BP 2 DAYS 1.80 1/10 10 54 BP 3 SWG	2.46 1/10 8 54 BP 3 SWG	2.46 1/10 8 54 BP 3 SWG
DAY: 2 SHIFT: DAYS	2.46 1/10 8 54 BP 3 SWG	3.13 1/10 10 54 BP 3 SWG	2.46 1/10 8 54 BP 3 SWG	2.46 1/10 8 54 BP 3 SWG
DAY: 2 SHIFT: SWING	2.46 1/10 8 54 BP 3 SWG	3.13 1/10 10 54 BP 3 SWG	2.46 1/10 8 54 BP 3 SWG	2.46 1/10 8 54 BP 3 SWG
DAY: 3 SHIFT: GYD	2.36 1/10 8 54 BP 3 SWG	1.93 1/10 10 54 BP 3 SWG	2.24 1/10 8 54 BP 3 SWG	2.24 1/10 8 54 BP 3 SWG
DAY: 3 SHIFT: DAYS				
DAY: 3 SHIFT: SWING				
DAY: 4 SHIFT: GYD				
DAY: 4 SHIFT: DAYS				

DAY: 4 SHIFT: SWING			
DAY: 5 SHIFT: GYD			
DAY: 5 SHIFT: DAYS			
DAY: 5 SHIFT: SWING			
DAY: 6 SHIFT: GYD			
0.73 1/10 9 54 BP 6 SWG			
DAY: 6 SHIFT: DAYS			
1.17 1/6 8 54 D 7 GYD	1.08 1/6 8 RW D 7 GYD		
DAY: 6 SHIFT: SWING			
0.29 1/10 8 54 ECLR 7 DAYS	1.73 1/6 8 RW D 7 GYD	0.64 1/6 8 54 D 7 DAYS 0.20 1/6 8 27 CP 7 SWG	
DAY: 7 SHIFT: GYD			
0.33 1/10 8 54 ECLR 7 DAYS 2.13 1/10 8 54 ECLR 7 SWG		3.36 1/6 8 27 CP 7 SWG	1.70 1/10 9 54 BP 7 SWG
DAY: 7 SHIFT: DAYS			
0.02 1/10 8 54 ECLR 7 SWG	0.29 1/6 10 27 CP 8 GYD	0.02 1/6 8 27 CP 7 SWG	1.79 1/6 10 27 CP 8 GYD
DAY: 7 SHIFT: SWING			

MACHINE:

DAY / SHIFT	17 QTY	THK	LEN	WID	GRADE	COMPL	18 QTY	THK	LEN	WID	GRADE	COMPL
DAY: 1 SHIFT: DAYS												
DAY: 1 SHIFT: SWING	0.39	1/10	8	54	BP	3 SWG						
DAY: 2 SHIFT: GYD	2.46	1/10	8	54	BP	3 SWG	0.14	1/10	8	54	BP	3 SWG
DAY: 2 SHIFT: DAYS	2.46	1/10	8	54	BP	3 SWG	2.46	1/10	8	54	BP	3 SWG
DAY: 2 SHIFT: SWING	2.46	1/10	8	54	BP	3 SWG	2.46	1/10	8	54	BP	3 SWG
DAY: 3 SHIFT: GYD	2.24	1/10	8	54	BP	3 SWG	2.24	1/10	8	54	BP	3 SWG
DAY: 3 SHIFT: DAYS												
DAY: 3 SHIFT: SWING												
DAY: 4 SHIFT: GYD												
DAY: 4 SHIFT: DAYS												

DAY: 4 SHIFT: SWING	
DAY: 5 SHIFT: GYD	
DAY: 5 SHIFT: DAYS	
	0.73 1/10 9 54 BP 5 SWG 0.98 1/10 8 54 BP 6 GYD
DAY: 5 SHIFT: SWING	
	0.75 1/10 9 54 BP 7 GYD
DAY: 6 SHIFT: GYD	
	2.12 1/10 9 54 BP 7 GYD
DAY: 6 SHIFT: DAYS	
0.41 1/6 8 RW D 6 SWG 0.62 1/10 8 54 ECLR 6 SWG 0.97 1/10 9 54 BP 7 SWG	1.93 1/10 9 54 BP 7 GYD
DAY: 6 SHIFT: SWING	
2.12 1/10 9 54 BP 7 SWG	
DAY: 7 SHIFT: GYD	
2.12 1/10 9 54 BP 7 SWG	
DAY: 7 SHIFT: DAYS	
0.56 1/10 9 54 BP 7 SWG	
DAY: 7 SHIFT: SWING	

JOINTER
OPERATION
MACHINE:

DAY / SHIFT	1						2						3						4					
	QTY	THK	LEN	WID	GRADE	COMPL	QTY	THK	LEN	WID	GRADE	COMPL	QTY	THK	LEN	WID	GRADE	COMPL	QTY	THK	LEN	WID	GRADE	COMPL
DAY: 2 SHIFT: SWING													6.08	1/6	8	27	CP	2 SWG	1.36	1/6	8	27	CP	3 DAYS
																			0.40	1/6	10	27	CP	3 DAYS
																			0.51	1/6	10	27	CP	3 DAYS
DAY: 3 SHIFT: GYD																								
DAY: 3 SHIFT: DAYS																								
DAY: 3 SHIFT: SWING							3.12	1/6	8	RW	D	4 SWG												
DAY: 4 SHIFT: GYD	2.32	1/6	8	RW	D	4 SWG													2.64	1/6	9	27	CP	4 SWG
	3.43	1/6	8	RW	D	5 GYD													1.85	1/6	9	27	CP	5 GYD
DAY: 4 SHIFT: DAYS	3.24	1/6	8	RW	D	5 GYD	3.32	1/6	8	RW	D	5 DAYS	7.37	1/6	9	27	CP	5 DAYS	4.46	1/6	9	27	CP	5 GYD
DAY: 4 SHIFT: SWING							3.18	1/6	8	RW	D	5 DAYS	0.08	1/6	9	27	CP	5 DAYS	0.09	1/6	8	RW	D	5 DAYS
DAY: 5 SHIFT: GYD																								
DAY: 5 SHIFT: DAYS																								
DAY: 5 SHIFT: SWING							2.99	1/6	8	RW	D	6 SWG												
DAY: 6 SHIFT: GYD	4.54	1/6	9	27	C	7 GYD	0.80	1/6	8	RW	D	6 SWG							0.43	1/6	8	RW	D	6 SWG
							2.81	1/6	8	RW	D	7 GYD												

DAY: 6 SHIFT: DAYS			
2.85 1/6 9 27 C 7 GYD			
DAY: 6 SHIFT: SWING			
		3.58 1/6 8 27 CP 7 DAYS	
DAY: 7 SHIFT: GYD			
5.58 7/32 8 RW D 8 GYD	2.71 7/32 8 RW D 8 GYD		2.64 1/6 9 27 C 7 SWG 1.79 1/6 10 27 CP 7 SWG
DAY: 7 SHIFT: DAYS			
1.88 7/32 8 RW D 8 GYD		4.44 1/6 10 27 C 8 GYD	0.29 1/6 10 27 CP 8 GYD
DAY: 7 SHIFT: SWING			

MACHINE:
5

DAY: 2 SHIFT: SWING

QTY	THK	LEN	WID	GRADE	COMPL
5.80	1/6	8	27	CP	3 GYD

DAY: 3 SHIFT: GYD

DAY: 3 SHIFT: DAYS

DAY: 3 SHIFT: SWING

DAY: 4 SHIFT: GYD

DAY: 4 SHIFT: DAYS

DAY: 4 SHIFT: SWING

DAY: 5 SHIFT: GYD

0.37	1/6	8	RW	D	5 SWG
1.14	1/6	9	27	CP	5 SWG

DAY: 5 SHIFT: DAYS

DAY: 5 SHIFT: SWING

1.14	1/6	9	27	CP	6 DAYS
1.52	1/6	9	27	C	6 SWG

DAY: 6 SHIFT: GYD

4.78	1/6	9	27	C	6 SWG

```
DAY:  6 SHIFT:    DAYS              |
------------------------------------|
                                    |
====================================|
                                    |
DAY:  6 SHIFT:  SWING               |
------------------------------------|
  1.57 1/6    8  27  CP     7 SWG   |
====================================|
                                    |
DAY:  7 SHIFT:     GYD              |
------------------------------------|
  0.52 1/6    8  27  CP     7 SWG   |
                                    |
====================================|
                                    |
DAY:  7 SHIFT:    DAYS              |
------------------------------------|
                                    |
====================================|
                                    |
DAY:  7 SHIFT:  SWING               |
------------------------------------|
==========================================================================================================================================
```

DRYER
OPERATION
MACHINE:

DAY / SHIFT	1	2	3	4
	QTY THK LEN WID LOG GRD COMPL	QTY THK LEN WID LOG GRD COMPL	QTY THK LEN WID LOG GRD COMPL	QTY THK LEN WID LOG GRD COMPL
DAY: 1 SHIFT: GYD				23.24 1/10 12 54 2M 1 DAYS
DAY: 1 SHIFT: DAYS				6.82 1/10 12 54 2M 1 DAYS 4.51 1/10 13 54 2M 1 DAYS
DAY: 1 SHIFT: SWING			21.02 1/10 8 54 3P 2 GYD	
DAY: 2 SHIFT: GYD			9.00 1/10 8 54 3P 2 GYD 6.92 1/10 8 54 3P 2 GYD 25.60 1/10 8 54 3P 2 DAYS	
DAY: 2 SHIFT: DAYS	3.08 1/6 8 27 3PSP 3 SWG	19.98 1/6 8 27 3PSP 2 SWG	4.41 1/10 8 54 3P 2 DAYS 11.25 1/10 8 RW 3M 2 DAYS 3.03 1/10 8 54 3P 2 DAYS	0.95 1/10 8 54 3P 2 SWG
DAY: 2 SHIFT: SWING	20.15 1/6 8 27 3PSP 2 SWG 3.00 7/32 8 RW 3M 2 SWG 5.46 1/6 8 37 3PSP 2 SWG 3.04 1/12 8 RW 3P 2 SWG 10.32 1/6 9 54 2M 3 GYD	4.31 1/6 8 27 3PSP 2 SWG		7.44 1/10 8 54 3P 2 SWG 3.00 1/6 12 27 3M 2 SWG 3.01 1/10 12 54 3M 2 SWG 6.59 1/10 12 RW 3M 2 SWG 3.03 1/10 8 54 3P 3 SWG 3.04 1/10 10 54 3P 2 SWG 6.23 1/10 10 54 SM 3 GYD
DAY: 3 SHIFT: GYD	2.29 1/6 9 54 2M 3 GYD			2.77 1/10 10 54 SM 3 GYD 6.92 1/10 8 54 3P 3 GYD 17.74 1/10 9 54 SM 3 DAYS
DAY: 3 SHIFT: DAYS		22.19 1/6 8 RW 2M 3 SWG	7.91 1/10 10 RW SM 3 DAYS 27.49 1/10 9 54 SM 3 SWG	12.29 1/10 9 54 SM 3 DAYS 13.08 1/10 10 54 3P 3 DAYS

Day / Shift				
DAY: 3 SHIFT: SWING	9.81 1/6 8 RW 2M 3 SWG 13.26 1/6 8 RW 3M 4 GYD	7.87 1/6 8 RW 2M 3 SWG 3.01 1/6 8 RW 3M 3 SWG 3.01 1/6 8 RW 3M 3 SWG 12.08 1/6 9 54 2M 3 SWG 16.44 1/6 8 RW 3M 4 GYD	2.53 1/10 9 54 3M 3 SWG 16.78 1/10 8 54 3P 3 SWG	
DAY: 4 SHIFT: GYD	8.31 1/6 8 RW 3M 4 GYD 5.30 1/6 8 RW 3M 4 GYD 3.01 1/6 8 RW 3M 4 GYD 13.79 1/6 8 RW 2M 4 GYD 11.97 1/6 8 RW 3PSP 4 DAYS	13.61 1/6 8 RW 3M 4 GYD 14.50 1/6 8 RW 2M 4 GYD 0.82 7/32 8 RW 3M 4 DAYS	8.87 1/6 10 54 2M 4 GYD 12.89 1/10 8 54 3P 4 GYD	
DAY: 4 SHIFT: DAYS	13.55 1/6 8 RW 3PSP 4 DAYS 3.97 1/6 9 54 2M 4 DAYS 16.97 1/6 8 RW 3M 4 SWG	12.01 7/32 8 RW 3M 4 DAYS 5.37 7/32 8 RW 3M 4 DAYS 15.17 1/6 9 54 2M 4 DAYS 10.34 1/6 8 RW 3M 4 SWG		18.20 1/10 9 54 3P 4 SWG
DAY: 4 SHIFT: SWING	4.69 1/6 8 RW 3M 4 SWG 3.01 1/6 8 RW 3M 4 SWG 12.73 1/6 8 RW 3M 4 SWG 10.51 1/6 9 37 3PSP 4 SWG	13.61 1/6 8 RW 3M 4 SWG	3.07 1/10 9 54 3P 4 SWG	11.83 1/10 9 54 3P 4 SWG
DAY: 5 SHIFT: GYD	8.20 1/6 9 54 2M 5 GYD	7.16 1/6 8 RW 3M 5 GYD	12.10 1/6 8 54 2M 5 GYD	
DAY: 5 SHIFT: DAYS		4.15 1/6 8 RW 3M 5 SWG	15.51 1/10 9 54 3P 5 SWG	
DAY: 5 SHIFT: SWING	11.01 7/32 8 RW 3M 5 SWG	13.61 1/6 8 RW 3M 5 SWG	11.83 1/10 9 54 3P 5 SWG	3.01 1/6 8 RW 3M 5 SWG 7.96 1/6 8 RW 3M 6 GYD
DAY: 6 SHIFT: GYD	3.08 1/6 8 54 2M 6 GYD 3.00 7/32 8 RW 3M 6 GYD 4.36 1/6 8 54 2M 6 GYD 5.26 1/6 8 54 2M 6 GYD	4.37 7/32 8 RW 3M 6 GYD 7.83 7/32 8 RW 3M 6 GYD 5.39 1/6 8 RW 3M 6 DAYS		13.61 1/6 8 RW 3M 6 GYD
DAY: 6 SHIFT: DAYS	0.97 1/6 8 RW 3M 6 SWG	13.61 1/6 8 RW 3M 6 DAYS	5.48 1/6 8 RW 3M 6 SWG	

DAY: 6 SHIFT: SWING			
2.03 1/6 8 RW 3M 6 SWG 13.00 1/6 8 54 2M 6 SWG		13.61 1/6 8 RW 3M 6 SWG	17.50 1/6 8 RW 3M 6 SWG
DAY: 7 SHIFT: GYD			
4.51 1/6 8 RW 3M 7 GYD	3.98 1/6 8 RW 3M 7 DAYS	9.86 1/6 8 27 3PSP 7 DAYS	
DAY: 7 SHIFT: DAYS			
	13.61 1/6 8 RW 3M 7 DAYS	4.49 1/6 8 27 3PSP 7 DAYS	8.36 1/6 8 27 3PSP 7 DAYS
DAY: 7 SHIFT: SWING			
6.90 1/6 8 RW 3M 7 SWG			
DAY: 8 SHIFT: GYD			
10.28 7/32 8 27 3M 8 GYD			
DAY: 8 SHIFT: DAYS			
DAY: 8 SHIFT: SWING			

MACHINE:

DAY / SHIFT	5 QTY	5 THK	5 LEN	5 WID	5 LOG GRD	5 COMPL	6 QTY	6 THK	6 LEN	6 WID	6 LOG GRD	6 COMPL	7 QTY	7 THK	7 LEN	7 WID	7 LOG GRD	7 COMPL
DAY: 1 SHIFT: GYD	18.73	1/10	12	54	2M	1 DAYS	4.51	1/10	12	54	2M	1 GYD						
							4.51	1/10	12	54	2M	1 GYD						
							18.73	1/10	12	54	2M	1 DAYS						
DAY: 1 SHIFT: DAYS	11.33	1/10	12	54	2M	1 DAYS	11.33	1/10	12	54	2M	1 DAYS	6.92	1/10	8	54	3P	1 DAYS
													15.48	1/10	8	54	3P	1 SWG
DAY: 1 SHIFT: SWING													14.53	1/10	8	54	3P	1 SWG
													6.92	1/10	8	54	3P	1 SWG
													16.05	1/10	10	54	SM	3 GYD
DAY: 2 SHIFT: GYD	0.09	1/10	8	54	3P	2 DAYS							13.98	1/10	10	54	SM	2 GYD
													24.05	1/10	8	54	3P	2 DAYS
DAY: 2 SHIFT: DAYS	29.84	1/10	8	54	3P	2 DAYS	13.77	1/10	10	54	SM	2 DAYS	2.94	1/10	8	54	3P	2 DAYS
	11.68	1/10	8	54	3P	2 SWG							3.03	1/10	8	54	3P	2 DAYS
													3.04	1/12	8	RW	3P	3 DAYS
													3.04	1/10	10	54	2M	2 DAYS
													3.04	1/10	10	54	2M	2 DAYS
DAY: 2 SHIFT: SWING	1.82	1/10	8	54	3P	2 SWG	13.49	1/10	8	54	3P	2 SWG	23.94	1/10	10	54	3P	3 GYD
	5.78	1/10	12	54	2M	2 SWG	3.01	1/10	12	54	2M	2 SWG						
	19.64	1/10	8	54	3P	3 GYD	6.88	1/10	12	54	2M	3 GYD						
DAY: 3 SHIFT: GYD	10.38	1/10	8	54	3P	3 GYD	0.06	1/10	12	54	2M	3 GYD	6.09	1/10	10	54	3P	3 GYD
	4.48	1/6	12	RW	2M	3 DAYS	16.78	1/10	8	54	3P	3 GYD						
							24.65	1/10	8	54	3P	3 DAYS						
DAY: 3 SHIFT: DAYS	9.07	1/6	12	RW	2M	3 DAYS	4.50	1/10	8	54	3P	3 DAYS	3.05	1/6	10	27	SMSP	3 DAYS
	2.53	1/10	9	54	SM	3 SWG	3.03	1/10	8	54	3P	3 DAYS						
							3.05	1/6	10	27	SMSP	3 DAYS						

DAY: 3 SHIFT: SWING	5.15 1/10 9 54 SM 3 SWG 6.68 1/10 9 54 SM 3 SWG	3.58 7/32 13 RW 3M 4 GYD	
DAY: 4 SHIFT: GYD		10.73 7/32 12 RW 3M 4 GYD	7.22 1/10 9 54 3P 4 GYD
DAY: 4 SHIFT: DAYS	3.00 1/6 13 RW 2M 4 DAYS 5.35 7/32 10 27 3M 4 DAYS		3.01 7/32 10 27 3M 4 DAYS
DAY: 4 SHIFT: SWING		25.28 1/6 9 27 3PSP 4 SWG	9.40 1/6 9 27 3PSP 5 GYD
DAY: 5 SHIFT: GYD	4.84 1/10 9 54 3P 5 GYD		20.38 1/6 9 27 3PSP 5 GYD
DAY: 5 SHIFT: DAYS			8.63 1/6 8 54 3M 5 DAYS 8.39 1/10 8 54 3P 5 DAYS
DAY: 5 SHIFT: SWING	3.08 1/6 8 54 3M 5 SWG 3.28 7/32 8 RW 3M 5 SWG 4.50 1/6 9 27 3PSP 5 SWG 6.05 1/6 8 54 2M 5 SWG 3.63 1/10 8 54 3P 5 SWG		3.08 1/6 8 54 3M 5 SWG
DAY: 6 SHIFT: GYD	3.81 1/10 8 54 3P 6 GYD 6.45 1/6 8 RW 3M 6 GYD		
DAY: 6 SHIFT: DAYS	4.50 1/6 9 27 3PSP 6 DAYS	13.00 1/6 8 54 2M 6 DAYS	

DAY: 6 SHIFT: SWING		
7.34 1/6 8 RW 3M 7 GYD		
DAY: 7 SHIFT: GYD		
13.61 1/6 8 RW 3M 7 GYD	11.14 1/6 8 RW 3M 7 GYD	5.75 1/6 8 54 2M 7 DAYS
DAY: 7 SHIFT: DAYS		
		7.34 1/6 8 54 2M 7 DAYS
DAY: 7 SHIFT: SWING		
7.17 1/6 10 27 SMSP 7 SWG	3.01 1/6 8 RW 3M 7 SWG	3.07 1/10 4 RW SM 7 SWG 12.83 1/10 4 RW SM 7 SWG 9.97 1/10 4 RW SM 7 SWG
DAY: 8 SHIFT: GYD		
	3.05 1/6 10 27 SMSP 8 GYD 13.44 1/6 8 RW 3M 8 GYD	
DAY: 8 SHIFT: DAYS		
DAY: 8 SHIFT: SWING		

APPENDIX C.2 TRIAL TWO RESULTS

TRIAL RUN #2

PROPOSED MILL SCHEDULE
FOR SPREADER OPERATION

MACHINE	PANEL SIZE	PANEL TYPE	PANEL GRADE	QUANTITY
1	4 X 8	SIDING	5/8 Premium Solid Center	17
1	4 X 8	SIDING	5/8 Premium Solid Center	15
1	4 X 8	SIDING	3/8 Sound	30
1	4 X 8	SIDING	3/8 Premium	15
1	4 X 8	SIDING	5/8 Premium C Center	50
1	4 X 8	SIDING	5/8 Sound Solid Center	40
1	4 X 8	SIDING	5/8 Sound C Center	45
1	4 X 8	SIDING	3/8 Premium	15
1	4 X 8	SIDING	3/8 Sound	35
1	4 X 8	SIDING	5/8 Premium C Center	65
2	4 X 9	SANDED	13/16 AC	35
2	4 X 12	SANDED	13/16 AC	12
2	4 X 9	SIDING	5/8 Sound C Center	55
2	4 X 9	SIDING	5/8 Premium C Center	40
2	4 X 9	SANDED	13/16 AC	50
2	4 X 9	SIDING	5/8 Sound Solid Center	40
2	4 X 9	SIDING	5/8 Sound C Center	55
2	4 X 9	SIDING	3/8 Sound	11
2	4 X 9	SIDING	5/8 Premium C Center	40
2	4 X 9	SIDING	5/8 Sound C Center	55
2	4 X 9	SIDING	5/8 Sound Solid	40
3	4 X 8	MARINE	9/16 AB	24
3	4 X 8	SHEATHING	5/8 CD	60
3	4 X 8	SANDED	13/16 AB	3
3	4 X 8	SANDED	5/16 AC	41
3	4 X 8	SANDED	1 1/16 AC	5.5
3	4 X 8	SANDED	13/16 AC	10
3	4 X 8	MARINE	7/16 AB	5
3	4 X 8	MARINE	9/16 AB	24
3	4 X 8	SHEATHING	5/8 CD	60
3	4 X 8	SANDED	7/16 AB	5
3	4 X 8	SANDED	11/16 AB	14
3	4 X 8	MARINE	13/16 AB	21
4	4 X 8	DURAPLY	3/8 G1S	25
4	4 X 8	DURAPLY	5/8 G2S	41
4	4 X 8	DURAPLY	9/16 G1S	15
4	4 X 8	DURAPLY	1/2 G1S	60
4	4 X 8	DURAPLY	3/8 G1S	25
4	4 X 8	DURAPLY	5/16 G1S	3
4	4 X 8	DURAPLY	1 G2S	17
4	4 X 8	DURAPLY	3/4 G2S	14
4	4 X 8	DURAPLY	1/2 G1S	40
6	4 X 10	DURAPLY	3/8 G1S	23
6	4 X 10	DURAPLY	3/8 G2S	10
6	4 X 10	DURAPLY	1/2 G2S	4
6	4 X 10	DURAPLY	1/4 G2S	6
6	4 X 9	DURAPLY	3/8 G1S	2
6	4 X 12	DURAPLY	1/2 G1S	4
6	4 X 10	DURAPLY	1/2 G1S	21
6	4 X 10	DURAPLY	3/4 G1S	3
6	4 X 10	SANDED	13/16 AC	23
6	4 X 10	SANDED	13/16 AB	10
6	4 X 10	SANDED	9/16 AC	31
6	4 X 10	SANDED	1 1/16 AC	9
6	4 X 10	SANDED	1 5/16 AC	1
6	~~4 X 10~~	~~SANDED~~	~~13/16 AC~~	~~46~~
6	4 X 12	SANDED	13/16 AC	2.5
6	4 X 12	MARINE	9/16 AA	3.5
6	4 X 12	MARINE	7/16 AA	2.5
6	4 X 12	SANDED	7/16 AC	2.5
6	4 X 12	SANDED	13/16 AC	53
6	4 X 12	SANDED	1 1/16 AC	2

TRIAL RUN #2

GREEN VENEER CODE SEQUENCE

NO.	CODE	NO.	CODE	NO.	CODE	NO.	CODE
1	12311	56	14242	111	15313	166	11453
2	12311	57	12341	112	11411	167	13353
3	12311	58	13331	113	11411	168	23242
4	15213	59	13331	114	15313		
5	15132	60	14213	115	15122		
6	14231	61	12311	116	12321		
7	13331	62	15143	117	15213		
8	13331	63	14242	118	15213		
9	15213	64	12341	119	12321		
10	12311	65	14343	120	15213		
11	12311	66	15122	121	12321		
12	15313	67	12321	122	15213		
13	14231	68	15313	123	15213		
14	13331	69	12311	124	12321		
15	15112	70	15313	125	15213		
16	14213	71	13331	126	15213		
17	15313	72	13331	127	12311		
18	15341	73	15313	128	12311		
19	15341	74	13331	129	15213		
20	15341	75	15313	130	12321		
21	14341	76	15112	131	15213		
22	14211	77	12311	132	12311		
23	12311	78	15313	133	15113		
24	15313	79	12311	134	14221		
25	15122	80	15213	135	15213		
26	13321	81	13331	136	15112		
27	12321	82	12311	137	15213		
28	15213	83	15143	138	15112		
29	12311	84	14242	139	12311		
30	15113	85	12341	140	13353		
31	14231	86	15313	141	12311		
32	13331	87	15112	142	12311		
33	14243	88	12311	143	13353		
34	12321	89	15113	144	22212		
35	15213	90	12311	145	12311		
36	13331	91	15113	146	11453		
37	13331	92	15213	147	12311		
38	13413	93	14213	148	12311		
39	12311	94	15142	149	15113		
40	15113	95	12341	150	14221		
41	14231	96	14243	151	11453		
42	13331	97	13331	152	12311		
43	13331	98	15313	153	12311		
44	14243	99	12311	154	13353		
45	12311	100	15313	155	22212		
46	15313	101	15122	156	12311		
47	12311	102	12321	157	13353		
48	13413	103	14213	158	22212		
49	13331	104	14213	159	12311		
50	13331	105	15132	160	15113		
51	15113	106	13331	161	14231		
52	14231	107	15213	162	13331		
53	15143	108	15313	163	15113		
54	14341	109	11411	164	14231		
55	15143	110	11411	165	13331		

TRIAL RUN #2

VENEER INVENTORY RESULTS

(IN UNIT LOADS)

VENEER GRADE				MINIMUM	MAXIMUM
7/32	8	C	27	0.00	12.67
7/32	8	C	RW	0.00	20.94
7/32	9	C	27	0.00	19.05
7/32	10	C	27	0.00	2.33
7/32	12	C	27	0.00	2.33
7/32	12	C	RW	0.00	28.15
1/6	8	C	54	0.00	6.64
1/6	8	C	27	0.00	51.00
1/6	8	C	RW	0.00	71.49
1/6	9	C	54	0.00	7.66
1/6	10	C	54	0.00	13.77
1/6	12	C	54	0.00	0.43
1/6	12	C	27	0.00	11.17
1/6	12	C	RW	0.00	6.71
1/10	8	C	54	0.00	93.14
1/10	8	C	RW	0.00	47.20
1/10	9	C	54	0.00	237.73
1/10	10	C	54	0.00	76.42
1/10	12	C	54	0.00	19.72
1/10	12	C	RW	0.00	5.35
1/12	8	C	54	0.00	4.64
1/12	8	C	RW	0.00	3.22
7/32	8	D	27	0.00	4.49
7/32	8	D	RW	0.00	20.49
7/32	9	D	27	0.00	8.19
7/32	10	D	27	0.00	1.20
7/32	12	D	27	0.00	0.60
7/32	12	D	RW	0.00	7.31
1/6	8	D	54	0.00	4.00
1/6	8	D	27	-0.01	4.44
1/6	8	D	RW	0.00	37.62
1/6	9	D	54	0.00	1.13
1/6	10	D	54	-0.06	1.91
1/6	12	D	54	0.00	0.62
1/6	12	D	27	-0.02	2.43
1/6	12	D	RW	0.00	2.22
1/10	8	D	54	0.00	35.59
1/10	8	D	RW	0.00	18.41
1/10	9	D	54	0.00	51.66
1/10	10	D	54	0.00	15.94
1/10	12	D	54	0.00	4.13
1/10	12	D	RW	0.00	1.78
1/12	8	D	54	0.00	1.31
1/12	8	D	RW	0.00	0.91
1/6	8	CP	54	0.00	7.34

TRIAL RUN #2

VENEER INVENTORY RESULTS

(IN UNIT LOADS)

VENEER GRADE				MINIMUM	MAXIMUM
1/6	8	CP	27	-0.02	7.27
1/6	10	CP	54	0.00	0.01
1/6	12	CP	54	0.00	0.53
1/6	12	CP	27	0.00	0.75
1/10	8	CP	54	-0.09	3.16
1/10	10	CP	54	0.00	1.26
1/10	12	CP	54	0.00	0.37
1/12	8	CP	54	0.00	0.05
1/6	8	BP	54	0.00	3.18
1/6	8	BP	RW	0.00	8.22
1/6	9	BP	54	0.00	6.58
1/6	10	BP	54	0.00	10.65
1/6	12	BP	27	0.00	5.83
1/6	12	BP	RW	0.00	2.84
1/10	8	BP	54	0.00	51.30
1/10	9	BP	54	0.00	95.18
1/10	10	BP	54	0.00	18.01
1/10	12	BP	54	0.00	9.40
1/10	12	BP	RW	0.00	1.78
1/12	8	BP	54	0.00	2.52
1/12	8	BP	RW	0.00	1.76
7/32	8	X	27	-0.01	0.49
7/32	8	X	RW	0.00	1.35
7/32	9	X	27	0.00	1.23
7/32	10	X	27	0.00	0.18
7/32	12	X	27	0.00	0.09
7/32	12	X	RW	0.00	1.10
1/6	8	X	54	0.00	2.70
1/6	8	X	RW	0.00	2.33
1/6	9	X	54	0.00	0.21
1/6	10	X	54	0.00	0.83
1/10	8	X	54	0.00	7.17
1/10	8	X	RW	0.00	2.00
1/10	9	X	54	0.00	19.31
1/10	10	X	54	0.00	4.78
1/10	12	X	54	-0.02	1.07
1/12	8	X	54	0.00	0.49
1/12	8	X	RW	0.00	0.18
1/10	8	DCLR	54	0.00	36.45
1/10	9	DCLR	54	0.00	74.84
1/10	10	DCLR	54	0.00	8.61
1/10	12	DCLR	54	0.00	3.66
1/12	8	DCLR	54	0.00	3.32
1/10	8	ECLR	54	0.00	43.23
1/10	9	ECLR	54	0.00	74.84

TRIAL RUN #2

VENEER INVENTORY RESULTS

<IN UNIT LOADS>

VENEER GRADE	MINIMUM	MAXIMUM
1/10 10 ECLR 54	0.00	8.61
1/10 12 ECLR 54	0.00	3.48
1/12 8 ECLR 54	0.00	2.99
1/6 8 A 54	-0.01	0.27
1/6 10 A 54	-0.01	0.18
1/10 8 A 54	0.00	28.19
1/10 9 A 54	0.00	130.06
1/10 10 A 54	0.00	9.16
1/10 12 A 54	-0.02	2.62
1/12 8 A 54	0.00	0.87
1/6 8 B 54	0.00	0.06
1/6 10 B 54	0.00	0.03
1/10 8 B 54	-0.02	3.21
1/10 9 B 54	0.00	3.95
1/10 10 B 54	-0.02	1.72
1/10 12 B 54	-0.01	0.48
1/12 8 B 54	0.00	0.16
1/10 8 M 54	-0.01	4.34
1/10 4 C RW	0.00	30.68
1/12 4 C RW	0.00	2.77
1/10 4 D RW	0.00	20.46
1/12 4 D RW	0.00	1.85
1/10 4 SOL RW	0.00	17.57
1/12 4 SOL RW	0.00	1.62

CBP-LEBANON
TASK NO. ________

SPREADER ORDER FILE 11-JUN-84

TASK NO.	RATE	PANEL GRADE	CONSTRUCTION METHOD	PRODUCTION QUANTITY
121143	2.08	8 SID PREM-SCR EXT 5/8	1 1 1 0 2 231 531 771 51 53	768
171121	4.17	8 SID SND EXT 3/8	1 1 0 0 1 331 531 51 51 52	3120
161121	4.17	8 SID PREMIUM EXT 3/8	1 1 0 0 1 231 531 51 51 52	1440
131143	2.08	8 SID PREM-CCR EXT 5/8	1 1 1 0 2 231 531 571 51 53	2760
141143	2.08	8 SID SND-SCTR EXT 5/8	1 1 1 0 2 331 531 771 51 53	960
151143	2.08	8 SID SND-CCTR EXT 5/8	1 1 1 0 2 331 531 511 51 53	1080
151242	2.08	9 SID SND-CCTR EXT 5/8	1 1 1 0 2 332 532 512 51 53	3300
131242	2.08	9 SID PREM-CCR EXT 5/8	1 1 1 0 2 232 532 572 51 53	1600
141242	1.85	9 SID SND-SCTR EXT 5/8	1 1 1 0 2 333 532 772 51 53	1600
171221	4.17	9 SID SND EXT 3/8	1 1 0 0 1 332 532 51 51 52	440
321131	2.08	8 MAR AB EXT 1/2	1 1 1 0 2 831 431 721 51 73	1920
222151	2.08	8 SAN AB INT 3/4	1 1 1 0 2 331 431 621 51 61	60
231111	4.17	8 SAN AC EXT 1/4	1 1 0 0 1 331 531 51 51 53	1640
231161	1.46	8 SAN AC EXT 1 0/0	1 1 1 1 3 331 531 521 511 52	110
231181	1.46	8 SAN AC EXT 1 1/4	1 1 2 0 3 321 521 521 51 51	44
231151	2.08	8 SAN AC EXT 3/4	1 1 1 0 2 331 531 521 51 51	200
321151	1.46	8 MAR AB EXT 3/4	1 1 2 0 3 831 431 721 51 73	280
321121	2.08	8 MAR AB EXT 3/8	1 1 1 0 2 831 431 731 51 74	100
321141	1.46	8 MAR AB EXT 5/8	1 1 2 0 3 831 431 731 51 73	280
231351	1.75	10 SAN AC EXT 3/4	1 1 1 0 2 333 533 523 51 51	1380
221351	1.75	10 SAN AB EXT 3/4	1 1 1 0 2 333 433 523 51 51	200
231331	1.75	10 SAN AC EXT 1/2	1 1 1 0 2 333 533 523 51 53	620
231361	1.25	10 SAN AC EXT 1 0/0	1 1 1 1 3 333 533 523 513 52	180
231381	1.25	10 SAN AC EXT 1 1/4	1 1 2 0 3 323 523 523 51 51	30
231451	1.46	12 SAN AC EXT 3/4	1 1 1 0 2 334 534 524 51 51	1380
231421	2.92	12 SAN AC EXT 3/8	1 1 0 0 1 334 534 51 51 51	100
231461	1.04	12 SAN AC EXT 1 0/0	1 1 1 1 3 334 534 524 514 53	40
311431	1.46	12 MAR AA EXT 1/2	1 1 1 0 2 834 834 724 51 73	70
311421	1.46	12 MAR AA EXT 3/8	1 1 1 0 2 834 834 734 51 74	50
231251	2.08	9 SAN AC EXT 3/4	1 1 1 0 2 332 532 522 51 51	1700
411121	3.75	8 DUR G1S EXT 3/8	1 1 0 0 1 431 531 51 51 52	2000
411131	1.87	8 DUR G1S EXT 1/2	1 1 1 0 2 431 531 531 51 53	2000
411111	3.75	8 DUR G1S EXT 1/4	1 1 0 0 1 431 531 51 51 54	120
411221	3.75	9 DUR G1S EXT 3/8	1 1 0 0 1 432 532 51 51 52	80
411321	3.17	10 DUR G1S EXT 3/8	1 1 0 0 1 433 533 51 51 52	920
411331	1.58	10 DUR G1S EXT 1/2	1 1 1 0 2 433 533 533 51 53	420
411351	1.58	10 DUR G1S EXT 3/4	1 1 1 0 2 433 533 513 51 52	60
411431	1.33	12 DUR G1S EXT 1/2	1 1 1 0 2 434 534 534 51 53	80
421141	1.87	8 DUR G2S EXT 5/8	1 1 1 0 2 431 431 531 51 52	820
421161	1.46	8 DUR G2S EXT 1 0/0	1 1 2 0 3 431 431 521 51 52	340
421321	1.58	10 DUR G2S EXT 3/8	1 1 0 0 1 433 433 51 51 52	400
431331	1.58	10 DUR G2S EXT 1/2	1 1 1 0 2 433 433 533 51 53	80
421311	1.58	10 DUR G2S EXT 1/4	1 1 0 0 1 433 433 51 51 54	340
461151	1.87	8 DUR CON G2S EXT 3/4	1 1 1 0 2 431 431 511 51 52	60
431151	1.46	8 DUR G2S-CCTR EXT 3/4	1 1 2 0 3 841 441 551 51 53	280
512142	1.87	8 SHE CDX INT 5/8	1 1 1 0 2 531 631 631 51 62	2400

CBP-LEBANON CURRENT INVENTORY 11-JUN-84

DRY INVENTORY:

GRADE	THK	LENGTH	WIDTH	QUANTITY

271

DRY-END SCHEDULE
CBP-LEBANON

11-JUN-84

SPREADER
OPERATION
MACHINE:

DAY / SHIFT	1	2	3	4
DAY: 4 SHIFT: GYD	QTY THK LEN GRADE CONST	QTY THK LEN GRADE CONST	QTY THK LEN GRADE CONST	QTY THK LEN GRADE CONST
	2001 3/8 8 SID SND EXT 1		180 1 0/0 10 SAN AC EXT 1	820 5/8 8 DUR G3S EXT 1
			588 1/2 10 SAN AC EXT 1	60 3/4 8 DUR CON G2S EXT 1
				13 1 0/0 8 DUR G2S EXT 1
DAY: 4 SHIFT: DAYS	1119 3/8 8 SID SND EXT 1	998 5/8 9 SID SND-CCIR EXT 2	32 1/2 10 SAN AC EXT 1	327 1 0/0 8 DUR G2S EXT 1
	883 1/4 8 SAN AC EXT 1		374` 3/4 10 SAN AC EXT 1	120 1/4 8 DUR G1S EXT 1
			200 3/4 10 SAN AB EXT 1	842 3/8 8 DUR G1S EXT I
			30 1 1/4 10 SAN AC EXT 1	
			100 3/8 12 SAN AC EXT 1	
			109 3/4 12 SAN AC EXT 1	
DAY: 4 SHIFT: SWING	757 1/4 8 SAN AC EXT 1	998 5/8 9 SID SND-CCIR EXT 2	700 3/4 12 SAN AC EXT 1	1158 3/8 8 DUR G1S EXT 1
	620 5/8 8 SID SND-CCIR EXT 3			330 1/3 8 DUR G1S EXT 1
DAY: 5 SHIFT: GYD	460 5/8 8 SID SND-CCIR EXT 3		571 3/4 12 SAN AC EXT 1	897 1/2 8 DUR G1S EXT 1
	200 3/4 8 SAN AC EXT 1		40 1 0/0 12 SAN AC EXT 1	
	44 1 1/4 8 SAN AC EXT 1			
	554 3/8 8 SID PREMIUM EXT 1			
DAY: 5 SHIFT: DAYS	886 3/8 8 SID PREMIUM EXT 1	998 5/8 9 SID SND-CCIR EXT 2		783 1/2 8 DUR G1S EXT 1
	556 5/8 8 SID PREM-CCR EXT 3			90 3/4 8 DUR G2S-CCIR EXT 1
DAY: 5 SHIFT: SWING	998 5/8 8 SID PREM-CCR EXT 3	306 5/8 9 SID SND-CCIR EXT 2		190 3/4 8 DUR G2S-CCIR EXT 1
		440 3/8 9 SID SND EXT 1		
		421 5/8 9 SID SND-SCIR EXT 2		
DAY: 6 SHIFT: GYD	998 5/8 8 SID PREM-CCR EXT 3			

DAY: 6 SHIFT: DAYS			
208 5/8 8 SID PREM-CCR EXT 3	888 5/8 9 SID SND-SCIR EXT 2		
768 5/8 8 SID PREM-SCR EXT 3			
23 5/8 8 SID SND-SCIR EXT 3			
DAY: 6 SHIFT: SWING			
937 5/8 8 SID SND-SCIR EXT 3	291 5/8 9 SID SND-SCIR EXT 2		
43 1 0/0 8 SAN AC EXT 1	672 3/4 9 SAN AC EXT 1		
DAY: 7 SHIFT: GYD			
67 1 0/0 8 SAN AC EXT 1			
60 3/4 8 SAN AB INT 1			
280 5/8 8 MAR AB EXT 1			
280 3/4 8 MAR AB EXT 1			
46 3/8 8 MAR AB EXT 1			
DAY: 7 SHIFT: DAYS			
54 3/8 8 MAR AB EXT 1	998 3/4 9 SAN AC EXT 1		
944 1/2 8 MAR AB EXT 1			
DAY: 7 SHIFT: SWING			
976 1/2 8 MAR AB EXT 1	30 3/4 9 SAN AC EXT 1		
30 5/8 8 SHE CDX INT 2	815 3/4 10 SAN AC EXT 1		
DAY: 8 SHIFT: GYD			
897 5/8 8 SHE CDX INT 2			
DAY: 8 SHIFT: DAYS			
897 5/8 8 SHE CDX INT 2	190 3/4 10 SAN AC EXT 1		
	50 3/8 12 MAR AA EXT 1		
	70 1/2 12 MAR AA EXT 1		
	601 5/8 9 SID PREM-CCR EXT 2		
DAY: 8 SHIFT: SWING			
586 5/8 8 SHE CDX INT 2	998 5/8 9 SID PREM-CCR EXT 2		

MACHINE:

DAY	SHIFT	5: QTY	5: THK	5: LEN	5: GRADE	5: CONST	6: QTY	6: THK	6: LEN	6: GRADE	6: CONST
4	GYD										
4	DAYS						80	1/2	12	DUR	G1S EXT 1
							80	3/8	9	DUR	G1S EXT 1
							400	3/8	10	DUR	G2S EXT 1
							229	1/4	10	DUR	G2S EXT 1
4	SWING						11	1/4	10	DUR	G2S EXT 1
							80	1/2	10	DUR	G2S EXT 1
							420	1/2	10	DUR	G1S EXT 1
							497	3/8	10	DUR	G1S EXT 1
5	GYD										
5	DAYS						423	3/8	10	DUR	G1S EXT 1
							60	3/4	10	DUR	G1S EXT 1
5	SWING										
6	GYD										
6	DAYS										

DAY: 6 SHIFT: SWING		
DAY: 7 SHIFT: GYD		
DAY: 7 SHIFT: DAYS		
DAY: 7 SHIFT: SWING		
DAY: 8 SHIFT: GYD		
DAY: 8 SHIFT: DAYS		
DAY: 8 SHIFT: SWING		

PATCHER
OPERATION
MACHINE:

DAY: 1 SHIFT: SWING

MACHINE	QTY	THK	LEN	WID	GRADE	COMPL
1	1.51	1/10	8	54	BP	3 GYD
2	1.51	1/10	8	54	BP	3 GYD
3						
4	7.37	1/10	8	54	BP	3 GYD

DAY: 2 SHIFT: GYD

MACHINE	QTY	THK	LEN	WID	GRADE	COMPL
1	2.46	1/10	8	54	BP	3 GYD
2	2.46	1/10	8	54	BP	3 GYD
3	0.91	1/10	10	54	BP	3 GYD
3	0.48	1/6	10	54	D	3 GYD
4	2.46	1/10	8	54	BP	3 GYD

DAY: 2 SHIFT: DAYS

MACHINE	QTY	THK	LEN	WID	GRADE	COMPL
1	0.17	1/10	8	54	BP	3 GYD
1	0.20	1/6	8	RW	D	3 DAYS
2	0.17	1/10	8	54	BP	3 GYD
3	0.14	1/6	10	54	D	3 GYD
3	0.61	1/6	12	RW	D	3 GYD
3	0.41	1/10	10	54	BP	3 GYD
3	1.23	1/10	8	54	BP	3 SWG
4	0.17	1/10	8	54	BP	3 GYD

DAY: 2 SHIFT: SWING

MACHINE	QTY	THK	LEN	WID	GRADE	COMPL
1	1.09	1/6	8	RW	D	3 DAYS
1	1.41	1/10	10	54	BP	3 SWG
2	0.81	1/6	8	54	D	3 DAYS
2	0.47	1/6	8	RW	D	3 DAYS
2	0.43	1/6	12	27	D	3 SWG
2	0.00	1/12	8	54	BP	3 SWG
3	2.46	1/10	8	54	BP	3 SWG
4						

DAY: 3 SHIFT: GYD

MACHINE	QTY	THK	LEN	WID	GRADE	COMPL
1	0.73	1/10	10	54	BP	3 SWG
1	1.38	1/10	9	54	BP	5 DAYS
2	0.37	1/12	8	54	BP	3 SWG
2	1.73	1/10	9	54	BP	4 GYD
3	0.85	1/10	8	54	BP	3 SWG
3	1.38	1/10	9	54	BP	5 DAYS
4						

DAY: 3 SHIFT: DAYS

MACHINE	QTY	THK	LEN	WID	GRADE	COMPL
1	2.12	1/10	9	54	BP	5 DAYS
2	0.39	1/10	9	54	BP	4 GYD
2	2.00	1/10	8	54	BP	4 DAYS
3	2.12	1/10	9	54	BP	5 DAYS
4						

DAY: 3 SHIFT: SWING

MACHINE	QTY	THK	LEN	WID	GRADE	COMPL
1	2.12	1/10	9	54	BP	5 DAYS
2	2.46	1/10	8	54	BP	5 SWG
3	2.13	1/10	9	54	BP	5 DAYS
4	2.46	1/10	8	54	BP	5 SWG

DAY: 4 SHIFT: GYD

MACHINE	QTY	THK	LEN	WID	GRADE	COMPL
1	2.12	1/10	9	54	BP	5 DAYS
2	2.46	1/10	8	54	BP	5 SWG
3	2.12	1/10	9	54	BP	5 DAYS
4	2.46	1/10	8	54	BP	5 SWG

DAY: 4 SHIFT: DAYS				
2.12 1/10 9 54 BP 5 DAYS	2.46 1/10 8 54 BP 5 SWG	2.12 1/10 9 54 BP 5 DAYS	2.46 1/10 8 54 BP 5 SWG	
DAY: 4 SHIFT: SWING				
0.15 1/10 9 54 BP 5 DAYS	2.46 1/10 8 54 BP 5 SWG	0.15 1/10 9 54 BP 5 DAYS	2.46 1/10 8 54 BP 5 SWG	
DAY: 5 SHIFT: GYD				
0.38 1/6 10 54 D 5 SWG 0.10 1/10 8 54 ECLR 6 GYD	0.17 1/10 8 54 BP 5 SWG		0.17 1/10 8 54 BP 5 SWG	
DAY: 5 SHIFT: DAYS				
0.38 1/10 8 54 ECLR 6 GYD 1.79 1/10 10 54 BP 6 SWG		0.23 1/10 8 54 BP 6 GYD 1.02 1/10 8 54 BP 6 DAYS		
DAY: 5 SHIFT: SWING				
2.12 1/10 10 54 BP 6 SWG		0.17 1/10 8 54 BP 6 DAYS		
DAY: 6 SHIFT: GYD				
0.21 1/10 10 54 BP 6 SWG				
DAY: 6 SHIFT: DAYS				
			0.81 1/10 10 54 BP 7 DAYS	
DAY: 6 SHIFT: SWING				
	0.44 1/10 12 54 ECLR 7 DAYS	0.44 1/10 12 54 ECLR 7 DAYS	0.15 1/10 10 54 BP 7 DAYS	

MACHINE: 5, 6, 7, 8

DAY	SHIFT	MACHINE	QTY	THK	LEN	WID	GRADE	COMPL
1	SWING	5	0.12	1/10	8	54	BP	2 SWG
1	SWING	5	0.26	1/10	10	54	BP	2 SWG
1	SWING	6	2.34	1/10	10	54	BP	2 SWG
1	SWING	7	0.08	1/10	8	54	BP	2 SWG
1	SWING	8	0.31	1/10	8	54	BP	2 DAYS
1	SWING	8	0.62	1/6	10	54	D	2 DAYS
1	SWING	8	0.50	1/10	10	54	BP	2 SWG
2	GYD	5	1.23	1/10	10	54	BP	3 SWG
2	GYD	5	0.89	1/10	9	54	BP	3 SWG
2	GYD	6	0.64	1/10	10	54	BP	3 SWG
2	GYD	6	1.72	1/10	8	54	BP	3 GYD
2	GYD	7	0.33	1/10	8	54	BP	2 SWG
2	GYD	7	1.01	1/10	10	54	BP	2 SWG
2	GYD	7	0.25	1/6	10	54	BP	2 SWG
2	GYD	7	0.06	1/10	10	54	BP	2 SWG
2	GYD	7	0.77	1/10	8	54	BP	3 SWG
2	GYD	8	0.67	1/10	10	54	BP	3 SWG
2	GYD	8	1.68	1/10	8	54	BP	3 DAYS
2	DAYS	5	2.12	1/10	9	54	BP	3 SWG
2	DAYS	6	0.60	1/10	8	54	BP	3 GYD
2	DAYS	6	0.14	1/12	8	54	BP	3 GYD
2	DAYS	6	1.45	1/10	9	54	BP	3 DAYS
2	DAYS	7	2.46	1/10	8	54	BP	3 SWG
2	DAYS	8	2.14	1/10	8	54	BP	3 DAYS
2	DAYS	8	0.37	1/10	9	54	BP	4 SWG
2	SWING	5	1.80	1/10	9	54	BP	3 SWG
2	SWING	5	0.32	1/10	9	54	BP	5 GYD
2	SWING	6	0.03	1/10	9	54	BP	3 DAYS
2	SWING	6	2.09	1/10	9	54	BP	4 GYD
2	SWING	7	2.46	1/10	8	54	BP	3 SWG
2	SWING	8	2.12	1/10	9	54	BP	4 SWG
3	GYD	5	2.12	1/10	9	54	BP	5 GYD
3	GYD	6	2.12	1/10	9	54	BP	4 GYD
3	GYD	7	0.17	1/10	8	54	BP	3 SWG
3	GYD	7	1.38	1/10	9	54	BP	5 DAYS
3	GYD	8	2.12	1/10	9	54	BP	4 SWG
3	DAYS	5	2.12	1/10	9	54	BP	5 GYD
3	DAYS	6	1.66	1/10	9	54	BP	4 GYD
3	DAYS	6	0.53	1/10	8	54	BP	4 DAYS
3	DAYS	7	2.12	1/10	9	54	BP	5 DAYS
3	DAYS	8	2.12	1/10	9	54	BP	4 SWG
3	SWING	5	2.12	1/10	9	54	BP	5 GYD
3	SWING	6	1.96	1/10	8	54	BP	4 DAYS
3	SWING	6	0.45	1/6	8	RW	D	4 SWG
3	SWING	6	0.21	1/6	8	RW	D	4 SWG
3	SWING	7	2.12	1/10	9	54	BP	5 DAYS
3	SWING	8	2.12	1/10	9	54	BP	4 SWG
4	GYD	5	2.12	1/10	9	54	BP	5 GYD
4	GYD	6	0.50	1/6	8	RW	D	4 SWG
4	GYD	6	2.08	1/10	8	54	BP	6 GYD
4	GYD	7	2.12	1/10	9	54	BP	5 DAYS
4	GYD	8	1.26	1/10	9	54	BP	4 SWG

DAY / SHIFT				
DAY: 4 SHIFT: DAYS	1.23 1/10 9 54 BP 5 GYD 0.41 1/6 9 54 D 5 GYD 0.73 1/10 8 54 ECLR 6 DAYS	2.46 1/10 8 54 BP 6 GYD	2.13 1/10 9 54 BP 5 DAYS	
DAY: 4 SHIFT: SWING	2.46 1/10 8 54 ECLR 6 DAYS	2.46 1/10 8 54 BP 6 GYD	0.15 1/10 9 54 BP 5 DAYS	0.31 1/10 8 54 BP 5 SWG 0.34 1/10 8 54 BP 6 GYD
DAY: 5 SHIFT: GYD	2.46 1/10 8 54 ECLR 6 DAYS	2.46 1/10 8 54 BP 6 GYD	3.26 1/6 8 27 CP 6 DAYS	2.46 1/10 8 54 BP 6 GYD
DAY: 5 SHIFT: DAYS	2.46 1/10 8 54 ECLR 6 DAYS	0.55 1/10 8 54 BP 6 GYD	3.36 1/6 8 27 CP 6 DAYS	0.55 1/10 8 54 BP 6 GYD
DAY: 5 SHIFT: SWING	0.30 1/10 8 54 ECLR 6 DAYS		0.41 1/6 8 27 CP 6 DAYS	
DAY: 6 SHIFT: GYD				
DAY: 6 SHIFT: DAYS				
DAY: 6 SHIFT: SWING				

MACHINE: 9, 10, 11, 12

DAY: 1 SHIFT: SWING

MACHINE	QTY	THK	LEN	WID	GRADE	COMPL
9	0.27	1/6	8	RW	D	2 DAYS
9	1.74	1/10	10	54	BP	2 SWG
10						
11	0.07	1/10	8	54	BP	2 DAYS
11	1.85	1/10	8	54	BP	3 GYD
12						

DAY: 2 SHIFT: GYD

MACHINE	QTY	THK	LEN	WID	GRADE	COMPL
9	0.29	1/10	10	54	BP	2 SWG
9	1.62	1/10	8	54	BP	2 SWG
9	0.43	1/10	9	54	BP	3 GYD
10	0.40	1/10	10	54	BP	2 SWG
10	0.07	1/10	8	54	BP	2 SWG
10	1.01	1/10	10	54	BP	3 GYD
10	0.06	1/10	10	54	BP	3 GYD
10	0.06	1/10	12	54	BP	3 SWG
11	2.46	1/10	8	54	BP	3 GYD
12	0.41	1/10	12	54	BP	3 GYD
12	0.32	1/10	8	54	BP	3 DAYS

DAY: 2 SHIFT: DAYS

MACHINE	QTY	THK	LEN	WID	GRADE	COMPL
9	0.97	1/10	9	54	BP	3 GYD
9	1.01	1/10	8	54	BP	3 DAYS
9	0.27	1/10	9	54	BP	4 SWG
10	1.68	1/10	12	54	BP	3 SWG
11	1.35	1/10	8	54	BP	3 GYD
11	0.66	1/12	8	54	BP	3 DAYS
11	0.27	1/10	9	54	BP	4 SWG
12	2.46	1/10	8	54	BP	3 DAYS

DAY: 2 SHIFT: SWING

MACHINE	QTY	THK	LEN	WID	GRADE	COMPL
9	2.12	1/10	9	54	BP	4 SWG
10	1.68	1/10	12	54	BP	3 SWG
11	2.12	1/10	9	54	BP	4 SWG
12	1.48	1/10	8	54	BP	3 DAYS
12	0.37	1/6	8	54	BP	3 DAYS
12	0.66	1/10	8	54	BP	4 GYD

DAY: 3 SHIFT: GYD

MACHINE	QTY	THK	LEN	WID	GRADE	COMPL
9	2.12	1/10	9	54	BP	4 SWG
10	0.12	1/10	12	54	BP	3 SWG
10	1.38	1/10	9	54	BP	5 DAYS
11	2.12	1/10	9	54	BP	4 SWG
12	2.46	1/10	8	54	BP	4 GYD

DAY: 3 SHIFT: DAYS

MACHINE	QTY	THK	LEN	WID	GRADE	COMPL
9	2.12	1/10	9	54	BP	4 SWG
10	2.12	1/10	9	54	BP	5 DAYS
11	2.12	1/10	9	54	BP	4 SWG
12	0.85	1/10	8	54	BP	4 GYD
12	1.38	1/10	9	54	BP	5 SWG

DAY: 3 SHIFT: SWING

MACHINE	QTY	THK	LEN	WID	GRADE	COMPL
9	2.12	1/10	9	54	BP	4 SWG
10	2.12	1/10	9	54	BP	5 DAYS
11	2.12	1/10	9	54	BP	4 SWG
12	2.12	1/10	9	54	BP	5 SWG

DAY: 4 SHIFT: GYD

MACHINE	QTY	THK	LEN	WID	GRADE	COMPL
9	1.26	1/10	9	54	BP	4 SWG
10	3.12	1/10	9	54	BP	5 DAYS
11	1.26	1/10	9	54	BP	4 SWG
12	2.12	1/10	9	54	BP	5 SWG

DAY: 4 SHIFT: DAYS			
	2.12 1/10 9 54 BP 5 DAYS		2.13 1/10 9 54 BP 5 SWG
DAY: 4 SHIFT: SWING			
	0.15 1/10 9 54 BP 5 DAYS		2.12 1/10 9 54 BP 5 SWG
DAY: 5 SHIFT: GYD			
2.06 1/6 8 27 CP 6 GYD	0.95 1/10 8 54 ECLR 6 GYD	1.93 1/10 8 54 ECLR 6 GYD	0.15 1/10 9 54 BP 5 SWG
DAY: 5 SHIFT: DAYS			
2.09 1/6 8 27 CP 6 GYD	1.53 1/10 8 54 ECLR 6 GYD	0.55 1/10 8 54 ECLR 6 GYD	
DAY: 5 SHIFT: SWING			
DAY: 6 SHIFT: GYD			
DAY: 6 SHIFT: DAYS			
			0.01 1/10 12 54 ECLR 7 DAYS
DAY: 6 SHIFT: SWING			
		0.52 1/6 12 27 CP 7 DAYS	0.62 1/10 12 54 ECLR 7 DAYS

MACHINE:

DAY: 1 SHIFT: SWING

Machine 13

DAY / SHIFT	QTY	THK	LEN	WID	GRADE	COMPL
DAY: 1 SHIFT: SWING						
DAY: 2 SHIFT: GYD	0.92	1/10	8	54	BP	3 GYD
DAY: 2 SHIFT: DAYS	0.74	1/10	8	54	BP	3 GYD
	0.55	1/10	12	54	BP	3 GYD
	0.92	1/10	8	54	BP	3 SWG
DAY: 2 SHIFT: SWING	2.33	1/10	8	54	BP	3 SWG
	0.30	1/6	8	RW	D	4 GYD
DAY: 3 SHIFT: GYD	3.25	1/6	8	RW	D	4 GYD
DAY: 3 SHIFT: DAYS	0.41	1/6	8	RW	D	4 GYD
	0.54	1/10	8	54	BP	4 GYD
	1.38	1/10	9	54	BP	5 SWG
DAY: 3 SHIFT: SWING	2.12	1/10	9	54	BP	5 SWG
DAY: 4 SHIFT: GYD	2.12	1/10	9	54	BP	5 SWG

Machine 14

DAY / SHIFT	QTY	THK	LEN	WID	GRADE	COMPL
DAY: 1 SHIFT: SWING						
DAY: 2 SHIFT: GYD	0.22	1/10	8	54	BP	2 SWG
	1.44	1/10	10	54	BP	3 GYD
DAY: 2 SHIFT: DAYS	0.59	1/10	10	54	BP	3 GYD
	1.53	1/10	10	54	BP	3 DAYS
DAY: 2 SHIFT: SWING	0.99	1/10	10	54	BP	3 DAYS
	1.73	1/6	8	RW	D	3 SWG
DAY: 3 SHIFT: GYD	0.12	1/6	8	RW	D	3 SWG
	0.65	1/10	9	54	BP	3 SWG
	1.38	1/10	9	54	BP	5 DAYS
DAY: 3 SHIFT: DAYS	2.12	1/10	9	54	BP	5 DAYS
DAY: 3 SHIFT: SWING	2.12	1/10	9	54	BP	5 DAYS
DAY: 4 SHIFT: GYD	2.12	1/10	9	54	BP	5 DAYS

Machine 15

DAY / SHIFT	QTY	THK	LEN	WID	GRADE	COMPL
DAY: 1 SHIFT: SWING						
DAY: 2 SHIFT: GYD	1.44	1/10	8	54	BP	3 GYD
	0.10	1/6	10	54	D	3 GYD
	0.13	1/10	12	54	BP	3 GYD
DAY: 2 SHIFT: DAYS	0.38	1/10	12	54	BP	3 GYD
	1.64	1/10	10	54	BP	3 DAYS
DAY: 2 SHIFT: SWING	0.49	1/10	10	54	BP	3 DAYS
	0.92	1/6	8	RW	D	3 DAYS
	0.31	1/10	10	54	BP	3 DAYS
	0.69	1/12	8	54	ECLR	3 SWG
DAY: 3 SHIFT: GYD	0.71	1/12	8	54	ECLR	3 SWG
	1.38	1/10	9	54	BP	5 DAYS
DAY: 3 SHIFT: DAYS	2.12	1/10	9	54	BP	5 DAYS
DAY: 3 SHIFT: SWING	2.12	1/10	9	54	BP	5 DAYS
DAY: 4 SHIFT: GYD	2.12	1/10	9	54	BP	5 DAYS

Machine 16

DAY / SHIFT	QTY	THK	LEN	WID	GRADE	COMPL
DAY: 1 SHIFT: SWING						
DAY: 2 SHIFT: GYD	0.34	1/6	10	54	D	2 SWG
	0.60	1/10	8	54	BP	3 GYD
	0.25	1/10	10	54	BP	3 GYD
DAY: 2 SHIFT: DAYS	0.91	1/10	10	54	BP	3 GYD
	0.41	1/10	10	54	BP	3 GYD
	0.63	1/10	12	54	BP	3 SWG
DAY: 2 SHIFT: SWING	1.68	1/10	12	54	BP	3 SWG
DAY: 3 SHIFT: GYD	0.58	1/10	12	54	BP	3 SWG
	1.38	1/10	9	54	BP	5 DAYS
DAY: 3 SHIFT: DAYS	2.12	1/10	9	54	BP	5 DAYS
DAY: 3 SHIFT: SWING	2.12	1/10	9	54	BP	5 DAYS
DAY: 4 SHIFT: GYD	2.12	1/10	9	54	BP	5 DAYS

DAY: 4 SHIFT: DAYS			
2.12 1/10 9 54 BP 5 SWG	2.12 1/10 9 54 BP 5 DAYS	2.12 1/10 9 54 BP 5 DAYS	2.12 1/10 9 54 BP 5 DAYS
DAY: 4 SHIFT: SWING			
2.12 1/10 9 54 BP 5 SWG	0.15 1/10 9 54 BP 5 DAYS	0.15 1/10 9 54 BP 5 DAYS 0.28 1/10 8 54 BP 5 DAYS 0.31 1/6 9 54 D 5 DAYS 1.14 1/10 8 54 ECLR 6 SWG	0.15 1/10 9 54 BP 5 DAYS
DAY: 5 SHIFT: GYD			
0.15 1/10 9 54 BP 5 SWG	2.02 1/10 8 54 BP 6 DAYS	2.46 1/10 8 54 ECLR 6 SWG	
DAY: 5 SHIFT: DAYS			
	2.46 1/10 8 54 BP 6 DAYS	2.46 1/10 8 54 ECLR 6 SWG	1.41 1/10 8 54 BP 6 GYD
DAY: 5 SHIFT: SWING			
	0.30 1/10 8 54 BP 6 DAYS	2.46 1/10 8 54 ECLR 6 SWG	
DAY: 6 SHIFT: GYD			
		0.17 1/10 8 54 ECLR 6 SWG	
DAY: 6 SHIFT: DAYS			
			0.89 1/6 10 54 D 7 GYD 0.01 1/10 12 54 ECLR 7 DAYS
DAY: 6 SHIFT: SWING			
			0.62 1/10 12 54 ECLR 7 DAYS

MACHINE:

DAY / SHIFT	17 QTY	THK	LEN	WID	GRADE	COMPL	18 QTY	THK	LEN	WID	GRADE	COMPL
DAY: 1 SHIFT: SWING							1.53	1/10	9	54	BP	3 GYD
DAY: 2 SHIFT: GYD	0.16	1/10	10	54	BP	2 SWG	2.12	1/10	9	54	BP	3 GYD
	0.39	1/10	9	54	BP	2 SWG						
	0.46	1/10	8	54	BP	3 DAYS						
DAY: 2 SHIFT: DAYS	2.46	1/10	8	54	BP	3 DAYS	1.16	1/10	9	54	BP	3 GYD
							0.66	1/12	8	54	ECLR	3 DAYS
							0.27	1/10	9	54	BP	4 SWG
DAY: 2 SHIFT: SWING	1.55	1/10	8	54	BP	3 DAYS	2.12	1/10	9	54	BP	4 SWG
DAY: 3 SHIFT: GYD	0.21	1/10	12	54	BP	3 SWG	2.12	1/10	9	54	BP	4 SWG
	1.13	1/10	9	54	BP	4 DAYS						
DAY: 3 SHIFT: DAYS	2.12	1/10	9	54	BP	4 DAYS	2.12	1/10	9	54	BP	4 SWG
DAY: 3 SHIFT: SWING	0.15	1/10	9	54	BP	4 DAYS	2.12	1/10	9	54	BP	4 SWG
	1.54	1/10	9	54	BP	5 SWG						
DAY: 4 SHIFT: GYD	2.12	1/10	9	54	BP	5 SWG	1.26	1/10	9	54	BP	4 SWG

DAY: 4 SHIFT: DAYS	
2.12 1/10 9 54 BP 5 SWG	
DAY: 4 SHIFT: SWING	
2.12 1/10 9 54 BP 5 SWG	
DAY: 5 SHIFT: GYD	
0.15 1/10 9 54 BP 5 SWG 0.33 1/10 8 54 BP 6 GYD 0.26 1/10 8 54 ECLR 6 GYD	1.41 1/10 8 54 BP 6 GYD 0.31 1/6 8 27 CP 6 SWG
DAY: 5 SHIFT: DAYS	
0.15 1/10 8 54 ECLR 6 GYD 2.31 1/10 8 54 BP 6 SWG	3.36 1/6 8 27 CP 6 SWG
DAY: 5 SHIFT: SWING	
2.46 1/10 8 54 BP 6 SWG	3.36 1/6 8 27 CP 6 SWG
DAY: 6 SHIFT: GYD	
0.17 1/10 8 54 BP 6 SWG	0.24 1/6 8 27 CP 6 SWG
DAY: 6 SHIFT: DAYS	
DAY: 6 SHIFT: SWING	

JOINTER
OPERATION
MACHINE:

DAY / SHIFT	1: QTY THK LEN WID GRADE COMPL	2: QTY THK LEN WID GRADE COMPL	3: QTY THK LEN WID GRADE COMPL	4: QTY THK LEN WID GRADE COMPL
DAY: 2 SHIFT: DAYS	0.28 1/6 8 RW D 2 DAYS		0.63 1/6 12 RW D 3 GYD	
DAY: 2 SHIFT: SWING		1.30 1/6 8 RW D 3 DAYS		
DAY: 3 SHIFT: GYD	0.44 1/6 12 27 D 3 DAYS		3.98 1/6 8 RW D 3 SWG	0.93 1/6 8 RW D 3 DAYS
DAY: 3 SHIFT: DAYS				
DAY: 3 SHIFT: SWING				
DAY: 4 SHIFT: GYD	0.46 1/6 8 RW D 4 DAYS			0.71 1/6 8 RW D 4 SWG
DAY: 4 SHIFT: DAYS				
DAY: 4 SHIFT: SWING				

DAY: 5 SHIFT: GYD			
DAY: 5 SHIFT: DAYS		7.39 1/6 8 27 CP 6 GYD	4.17 1/6 8 27 CP 5 SWG
DAY: 5 SHIFT: SWING	0.46 1/6 8 RW D 6 GYD		
DAY: 6 SHIFT: GYD			
DAY: 6 SHIFT: DAYS			
DAY: 6 SHIFT: SWING			
DAY: 7 SHIFT: GYD		0.53 1/6 12 27 CP 7 DAYS	
DAY: 7 SHIFT: DAYS			
DAY: 7 SHIFT: SWING			

MACHINE: 5

DAY: 2 SHIFT: DAYS

QTY	THK	LEN	WID	GRADE	COMPL

DAY: 2 SHIFT: SWING

DAY: 3 SHIFT: GYD

QTY	THK	LEN	WID	GRADE	COMPL
0.49	1/6	8	RW	D	3 DAYS
1.85	1/6	8	RW	D	3 DAYS

DAY: 3 SHIFT: DAYS

DAY: 3 SHIFT: SWING

DAY: 4 SHIFT: GYD

DAY: 4 SHIFT: DAYS

DAY: 4 SHIFT: SWING

DAY: 5 SHIFT: GYD

7.04 1/6 8 27 CP 5 SWG

DAY: 5 SHIFT: DAYS

DAY: 5 SHIFT: SWING

DAY: 6 SHIFT: GYD

DAY: 6 SHIFT: DAYS

DAY: 6 SHIFT: SWING

DAY: 7 SHIFT: GYD

DAY: 7 SHIFT: DAYS

DAY: 7 SHIFT: SWING

DRYER
OPERATION
MACHINE:

DAY	SHIFT	MACHINE	QTY	THK	LEN	WID	LOG GRD	COMPL
1	GYD	4	15.31	1/10	8	54	3P	1 GYD
1	GYD	4	3.53	1/10	10	54	SM	1 DAYS
1	DAYS	3	6.92	1/10	8	54	3P	1 DAYS
1	DAYS	3	9.52	1/10	8	54	3P	1 DAYS
1	DAYS	3	1.47	1/10	8	54	3P	1 SWG
1	DAYS	4	12.80	1/10	10	54	SM	1 DAYS
1	DAYS	4	20.93	1/10	8	54	3P	1 DAYS
1	DAYS	4	3.12	1/6	12	RW	2M	1 SWG
1	SWING	3	14.27	1/10	8	54	3P	1 SWG
1	SWING	3	21.63	1/10	8	54	3P	1 SWG
1	SWING	3	5.36	1/10	8	54	3P	1 SWG
1	SWING	3	0.31	1/10	10	54	SM	2 GYD
1	SWING	4	3.71	1/6	12	RW	2M	1 SWG
1	SWING	4	3.01	1/6	10	54	3M	1 SWG
1	SWING	4	3.02	7/32	12	RW	3M	1 SWG
1	SWING	4	3.02	7/32	12	RW	3M	1 SWG
1	SWING	4	9.07	1/6	12	27	2M	1 SWG
1	SWING	4	8.10	1/10	10	54	SM	2 GYD
2	GYD	2	9.02	1/6	8	RW	3M	2 DAYS
2	GYD	3	8.03	1/10	10	54	SM	2 GYD
2	GYD	3	3.07	1/10	8	RW	3M	2 GYD
2	GYD	3	12.78	1/10	8	RW	3M	2 GYD
2	GYD	3	5.61	1/10	10	54	SM	2 GYD
2	GYD	3	3.03	1/10	8	54	3P	2 GYD
2	GYD	3	4.50	1/10	8	54	3P	2 DAYS
2	GYD	4	5.74	1/10	10	54	SM	2 GYD
2	GYD	4	30.03	1/10	9	54	3P	2 GYD
2	GYD	4	0.32	1/10	12	RW	2M	2 DAYS
2	DAYS	1	6.54	1/6	8	RW	3M	2 DAYS
2	DAYS	2	31.04	1/6	8	RW	3M	2 DAYS
2	DAYS	2	20.16	1/6	8	RW	3M	2 DAYS
2	DAYS	2	1.09	1/12	8	RW	SM	2 SWG
2	DAYS	3	7.09	1/10	8	54	3P	2 DAYS
2	DAYS	3	30.03	1/10	9	54	3P	2 DAYS
2	DAYS	3	0.48	1/10	10	54	SM	2 SWG
2	DAYS	4	8.59	1/10	12	RW	2M	2 DAYS
2	DAYS	4	6.96	1/10	8	RW	3M	2 DAYS
2	DAYS	4	12.60	1/10	9	54	3P	2 DAYS
2	DAYS	4	4.18	1/6	12	27	2M	2 SWG
2	SWING	1	4.77	1/6	8	RW	3M	2 SWG
2	SWING	1	3.00	7/32	8	27	3M	2 SWG
2	SWING	1	3.01	1/6	8	RW	2M	2 SWG
2	SWING	1	8.83	1/6	8	54	2M	2 SWG
2	SWING	1	12.36	7/32	9	27	3M	2 SWG
2	SWING	1	0.97	1/6	8	RW	3M	3 GYD
2	SWING	2	1.95	1/12	8	RW	SM	2 SWG
2	SWING	2	15.12	1/6	8	RW	2M	2 SWG
2	SWING	2	7.74	7/32	8	27	3M	2 SWG
2	SWING	2	4.00	7/32	8	RW	3M	2 SWG
2	SWING	2	3.00	7/32	8	RW	3M	2 SWG
2	SWING	2	3.01	1/6	8	RW	2M	2 SWG
2	SWING	2	7.78	1/6	8	RW	2M	3 GYD
2	SWING	3	2.56	1/10	10	54	SM	2 SWG
2	SWING	3	4.37	1/10	8	RW	3M	2 SWG
2	SWING	3	3.07	1/10	8	RW	3M	2 SWG
2	SWING	3	8.65	1/10	8	54	3P	2 SWG
2	SWING	3	3.81	1/10	8	54	3P	2 SWG
2	SWING	3	1.15	1/10	9	54	3P	3 GYD
2	SWING	4	3.18	1/6	12	27	2M	2 SWG
2	SWING	4	3.01	1/10	12	54	3P	2 SWG
2	SWING	4	28.72	1/10	9	54	3P	3 GYD

DAY: 3 SHIFT: GYD			
3.89 1/6 8 RW 3M 3 GYD	3.80 1/6 8 RW 3M 3 GYD	5.76 1/10 9 54 3P 3 GYD	1.31 1/10 9 54 3P 3 GYD
3.04 1/12 8 RW SM 3 GYD	3.01 1/6 8 RW 3M 3 GYD	6.91 1/10 9 54 3P 3 GYD	6.91 1/10 9 54 3P 3 GYD
5.73 7/32 8 RW 3M 3 GYD	3.29 1/12 8 54 1P 3 GYD	24.19 1/10 9 54 3P 3 DAYS	6.91 1/10 9 54 3P 3 GYD
3.00 7/32 8 RW 3M 3 GYD	3.06 1/13 8 54 1P 3 GYD		6.91 1/10 9 54 3P 3 GYD
12.36 7/32 9 27 3M 3 GYD	3.06 1/12 8 54 1P 3 GYD		14.82 1/10 9 54 3P 3 DAYS
6.01 1/6 8 RW 3M 3 GYD			
5.73 7/32 8 27 3M 3 GYD			
1.15 1/6 8 RW 3M 3 DAYS			
DAY: 3 SHIFT: DAYS			
2.65 1/6 8 RW 3M 3 DAYS		5.84 1/10 9 54 3P 3 DAYS	6.91 1/10 9 54 3P 3 DAYS
12.36 7/32 9 27 3M 3 DAYS		11.76 1/10 10 54 SM 3 DAYS	
10.78 1/6 8 RW 3M 3 DAYS		10.75 1/10 9 54 3P 3 SWG	
6.95 1/12 8 54 1P 3 DAYS			
DAY: 3 SHIFT: SWING			
		19.28 1/10 9 54 3P 3 SWG	
DAY: 4 SHIFT: GYD			
		12.14 1/10 8 RW 3M 4 GYD	13.31 1/10 9 54 3P 4 DAYS
		22.66 1/10 8 54 3P 4 DAYS	
DAY: 4 SHIFT: DAYS			
4.15 1/6 8 RW 3M 4 DAYS	3.86 7/33 9 27 3M 4 DAYS	7.35 1/10 8 54 3P 4 DAYS	16.51 1/10 9 54 3P 4 DAYS
9.02 1/6 8 RW 3M 4 DAYS	6.10 1/6 8 RW 3M 4 SWG		
8.75 1/6 8 RW 3M 4 SWG			
DAY: 4 SHIFT: SWING			
10.17 1/6 8 RW 3M 4 SWG	10.52 1/6 8 RW 3M 4 SWG	6.92 1/10 8 54 3P 4 SWG	
3.36 1/6 8 RW 3M 4 SWG			
6.19 1/6 8 RW 3M 4 SWG			
22.62 1/6 8 27 3PSP 5 GYD			
DAY: 5 SHIFT: GYD			
5.54 1/6 8 27 3PSP 5 GYD	18.56 1/6 8 RW 3M 5 GYD		
17.86 1/6 8 RW 3M 5 GYD	11.05 1/6 8 RW 3M 5 DAYS		
8.47 1/6 9 54 3M 5 GYD			
DAY: 5 SHIFT: DAYS			
	3.62 1/6 8 RW 3M 5 DAYS	5.28 1/10 8 54 3P 5 DAYS	12.37 1/10 8 54 3P 5 DAYS
	18.47 7/32 8 RW 3M 5 DAYS	3.03 1/10 8 54 3P 5 DAYS	2.94 1/10 8 54 3P 5 SWG
	16.63 1/6 8 37 3PSP 5 DAYS	3.03 1/10 8 54 3P 5 DAYS	
	4.14 1/6 8 27 3PSP 5 SWG		

DAY: 5 SHIFT: SWING			
	34.99 1/6 8 27 3PSP 5 SWG	11.63 1/10 10 54 SM 6 GYD	14.70 1/10 8 54 3P 5 SWG
DAY: 6 SHIFT: GYD			
3.00 7/32 8 27 3M 6 GYD 3.00 7/32 8 27 3M 6 GYD 3.08 1/12 4 RW 1P 6 GYD 16.02 7/32 8 RW 3M 6 DAYS	3.01 1/6 8 RW 3M 6 GYD 3.01 1/6 8 RW 3M 6 GYD 4.41 1/6 9 54 2M 6 DAYS	11.00 1/10 10 54 SM 6 GYD	
DAY: 6 SHIFT: DAYS			
10.83 7/32 8 RW 3M 6 DAYS 12.92 7/32 8 RW 3M 6 SWG	7.67 1/6 9 54 2M 6 DAYS 3.08 1/12 4 RW 1P 6 DAYS		
DAY: 6 SHIFT: SWING			
12.10 7/32 8 RW 3M 6 SWG		9.82 1/6 10 54 2M 6 SWG	
DAY: 7 SHIFT: GYD			
DAY: 7 SHIFT: DAYS			
	4.91 7/32 8 RW 3M 7 DAYS 3.08 1/12 4 RW 1P 7 DAYS		3.01 1/6 12 27 SMSP 7 DAYS
DAY: 7 SHIFT: SWING			

MACHINE:

	5							6							7						
	QTY	THK	LEN	WID	LOG	GRD	COMPL	QTY	THK	LEN	WID	LOG	GRD	COMPL	QTY	THK	LEN	WID	LOG	GRD	COMPL
DAY: 1 SHIFT: GYD															30.02	1/10	8	54	3P		1 GYD
															3.01	1/6	10	54	2M		1 GYD
															3.04	1/10	10	54	SM		1 GYD
															3.53	1/10	10	54	SM		1 DAYS
DAY: 1 SHIFT: DAYS								4.07	1/10	9	54	3P		1 SWG	7.61	1/10	10	54	SM		1 DAYS
															4.75	1/6	10	54	2M		1 DAYS
															3.07	1/10	9	54	3P		1 DAYS
															6.44	1/10	10	54	SM		1 DAYS
															3.07	1/10	8	RW	3M		1 DAYS
															3.04	1/10	10	54	SM		1 DAYS
															7.30	1/10	9	54	3P		1 SWG
DAY: 1 SHIFT: SWING	15.31	1/10	8	54	3P		1 SWG	13.67	1/10	9	54	3P		1 SWG	10.44	1/10	9	54	3P		1 SWG
	7.44	1/6	10	54	2M		1 SWG	5.05	1/10	10	54	SM		1 SWG	3.04	1/10	10	54	SM		1 SWG
	3.01	1/10	12	54	3M		1 SWG	3.01	7/32	10	27	3M		1 SWG	14.19	1/10	8	54	3P		1 SWG
	4.22	1/10	9	54	SM		2 GYD	3.03	1/10	8	54	3P		1 SWG	6.91	1/10	9	54	3P		1 SWG
								5.43	1/10	8	RW	3M		1 SWG	3.53	1/10	9	54	3P		2 GYD
								5.67	1/10	10	54	SM		2 GYD							
DAY: 2 SHIFT: GYD	3.46	1/10	9	54	SM		2 GYD	5.47	1/10	10	54	SM		2 GYD	26.50	1/10	9	54	3P		2 GYD
	11.47	7/32	8	RW	3M		2 GYD	3.01	1/10	12	54	2M		2 GYD	5.97	1/10	8	54	3P		2 GYD
	5.61	1/10	10	54	SM		2 GYD	16.52	1/10	8	54	3P		2 GYD	5.71	1/10	8	54	3P		2 DAYS
	3.00	1/6	12	RW	2M		2 GYD	9.13	1/10	12	54	3P		2 DAYS							
	6.44	1/10	10	54	SM		2 GYD														
	3.01	1/6	10	54	2M		2 GYD														
	2.66	1/10	13	54	2M		2 DAYS														
DAY: 2 SHIFT: DAYS	0.35	1/10	12	54	2M		2 DAYS	3.93	1/10	12	54	3P		2 DAYS	8.91	1/10	8	54	3P		2 DAYS
	16.78	1/10	8	54	3P		2 DAYS	3.01	1/10	12	54	3M		2 DAYS	6.91	1/10	9	54	3P		2 DAYS
	3.00	1/6	12	27	2M		2 DAYS	11.69	1/10	10	54	SM		2 DAYS	6.91	1/10	9	54	3P		2 DAYS
	6.91	1/10	9	54	3P		2 DAYS	3.01	1/10	12	54	3P		2 DAYS	15.13	1/10	9	54	3P		2 SWG
	10.68	1/10	9	54	3P		2 SWG	6.91	1/10	9	54	3P		2 DAYS							
								3.76	1/10	9	54	3P		2 SWG							
DAY: 2 SHIFT: SWING	19.35	1/10	9	54	3P		2 SWG	3.15	1/10	9	54	3P		2 SWG	14.90	1/10	9	54	3P		2 SWG
	10.69	1/10	12	54	3P		2 SWG	3.04	1/10	10	54	SM		2 SWG	14.08	1/10	8	RW	3M		2 SWG
	2.40	7/32	12	RW	3M		3 GYD	30.03	1/10	9	54	3P		2 SWG	7.85	1/10	8	RW	3M		2 SWG
								0.31	1/10	9	54	3P		3 GYD	1.15	1/10	9	54	3P		3 GYD

DAY: 3 SHIFT: GYD		
14.81 7/32 12 RW 3M 3 GYD 5.99 1/10 8 RW 3M 3 GYD 10.83 1/10 9 54 3P 3 DAYS	6.60 1/10 9 54 3P 3 GYD 30.03 1/10 9 54 3P 3 GYD 0.17 7/32 12 RW 3M 3 DAYS	5.76 1/10 9 54 3P 3 GYD 6.91 1/10 9 54 3P 3 GYD 5.45 1/10 9 54 3P 3 GYD
DAY: 3 SHIFT: DAYS		
19.20 1/10 9 54 3P 3 DAYS 3.07 1/10 8 RW 3M 3 DAYS 3.04 1/10 10 54 5M 3 DAYS 10.75 1/10 9 54 3P 3 SWG	13.14 7/32 12 RW 3M 3 DAYS 3.02 7/32 12 27 3M 3 DAYS 4.36 1/6 12 RW 2M 3 DAYS 7.83 1/10 9 54 3P 3 DAYS 1.21 1/10 8 RW 3M 3 SWG	
DAY: 3 SHIFT: SWING		
19.28 1/10 9 54 3P 3 SWG 10.75 1/10 9 54 3P 4 GYD	1.86 1/10 8 RW 3M 3 SWG 3.24 1/10 8 RW 3M 3 SWG 3.07 1/10 9 54 3P 3 SWG 10.75 1/10 9 54 3P 4 GYD	
DAY: 4 SHIFT: GYD		
19.28 1/10 9 54 3P 4 GYD 3.01 7/32 10 27 3M 4 GYD 3.88 1/10 8 RW 3M 4 GYD 9.26 1/10 8 54 3P 4 GYD	19.28 1/10 9 54 3P 4 GYD	7.44 1/10 8 54 3P 4 GYD 3.03 1/10 8 54 3P 4 GYD 6.92 1/10 8 54 3P 4 GYD 32.66 1/10 8 54 3P 4 DAYS
DAY: 4 SHIFT: DAYS		
	6.92 1/10 8 54 3P 4 DAYS 16.26 1/10 8 54 3P 4 SWG	7.35 1/10 8 54 3P 4 DAYS
DAY: 4 SHIFT: SWING		
	13.75 1/10 8 54 3P 4 SWG	
DAY: 5 SHIFT: GYD		

DAY: 5 SHIFT: DAYS		
	3.03 1/10 8 54 3P 5 DAYS	
DAY: 5 SHIFT: SWING		
5.28 1/10 8 54 3P 5 SWG 18.25 1/10 8 54 3P 5 SWG 4.41 1/10 8 54 3P 5 SWG 3.03 1/10 8 54 3P 5 SWG		
DAY: 6 SHIFT: GYD		
		15.02 1/10 4 RW SM 6 GYD 15.02 1/10 4 RW SM 6 GYD 17.86 1/10 4 RW SM 6 DAYS
DAY: 6 SHIFT: DAYS		
		12.17 1/10 4 RW SM 6 DAYS 3.63 1/10 4 RW SM 6 DAYS 21.81 1/10 4 RW SM 6 SWG
DAY: 6 SHIFT: SWING		
		8.22 1/10 4 RW SM 6 SWG 4.82 1/10 4 RW SM 6 SWG
DAY: 7 SHIFT: GYD		
	5.36 1/10 10 54 SM 7 GYD	
DAY: 7 SHIFT: DAYS		
	3.01 1/6 10 54 2M 7 DAYS	3.73 1/10 4 RW SM 7 DAYS
DAY: 7 SHIFT: SWING		

APPENDIX C.3 INVENTORY VERIFICATION PROGRAMS

```
C
C     JULY 9, 1984
C
C     THIS PROGRAM TRANSFORMS EACH TASK ASSIGNED TO THE DRYER OPERATION
C     INTO AN INVENTORY UPDATE EVENT.
C
C     AUTHOR:  ERIC SCHILD
C
      PROGRAM DTASK
      INTEGER NT,NR,CD,CODE,COMP,START,PRAT,MACH,N(720),GRD(720),
     *THK(720),LEN(720),WID(720),INDEX
      REAL PRATE,QTY
      NT=4
      NR=1
C
C     READING THE GREEN VENEER GRADE CODE
C
      CALL ASSIGN(1,'SY:DRYCOD.DAT')
      INDEX=1
5     READ(1,10,END=15) N(INDEX),GRD(INDEX),THK(INDEX),LEN(INDEX),
     *WID(INDEX)
10    FORMAT(T2,2I2,3I1)
      INDEX=INDEX+1
      GO TO 5
15    CONTINUE
      CALL CLOSE(1)
C
C     READING THE TASK ASSIGNED TO THE DRYER OPERATION
C
      CALL ASSIGN(1,'SY:DRYTSK.DAT')
      CALL ASSIGN(2,'SY:DRYTK1.DAT')
20    READ(1,25,END=35) COMP,START,CODE,PRAT,MACH
25    FORMAT(T2,2I6,I4,I6,I3)
C
C     DRYER PRODUCTION RATE
C
      PRATE=FLOAT(PRAT)/100000.
C
C     TASK QUANTITY
C
      QTY=FLOAT(COMP-START)*PRATE
C
C     TRANSFORMING GREEN VENEER CODES TO SINGLE VALUE
C
      CD=THK(CODE)*100+LEN(CODE)*10+WID(CODE)
C
C     WRITING INVENTORY UPDATE FILE FOR DRYER OPERATION
C
      WRITE(2,30) COMP,QTY,CODE,CD,NT,NR
30    FORMAT(T2,I6,F6.2,2I4,2I2)
      GO TO 20
35    CONTINUE
      CALL CLOSE(1)
      CALL CLOSE(2)
      STOP
      END
```

```
C
C    JULY 8. 1984
C
C    THIS PROGRAM TRANSFORMS THE TASKS ASSIGNED TO THE JOINTER AND PATCHER
C    OPERATIONS TO INVENTORY UPDATE EVENTS.
C
C    AUTHOR:  ERIC SCHILD
C
      PROGRAM JTASK
      INTEGER NR(2).NT(2).INDEX.GRD(660).THK(660).LEN(660).WID(660).
     *CODE.DINVEN(11.4.5.3),G,T.L.W.NPATCH(9,4,5,3).COMP,START.
     *PRAT.MACH.COD.PCODE.CD
      REAL PRATE.QTY
      DATA NT.NR/2.3.0.1/
C
C    READING THE DRY VENEER GRADE CODE
C
      CALL ASSIGN(1.'SY:INVCOD.DAT')
      INDEX=1
5     READ(1.10.END=15) GRD(INDEX).THK(INDEX).LEN(INDEX).WID(INDEX)
10    FORMAT(T2.I2.3I1)
      DINVEN(GRD(INDEX).THK(INDEX).LEN(INDEX).WID(INDEX))=INDEX
      INDEX=INDEX+1
      GO TO 5
15    CONTINUE
      CALL CLOSE(1)
C
C    READING THE PATCHER DEVELOPMENT CODE FILE
C
      CALL ASSIGN(1.'SY:PATCOD.DAT')
      INDEX=1
20    READ(1.25.END=30) G.T.L.W
25    FORMAT(T2.I2.3I1)
      NPATCH(G.T.L.W)=INDEX
      INDEX=INDEX+1
      GO TO 20
30    CONTINUE
      CALL CLOSE(1)
C
C    READING THE TASK ASSIGNED TO THE PATCHER OPERATION
C
      CALL ASSIGN(1.'SY:PATTSK.DAT')
      CALL ASSIGN(2.'SY:PATTK1.DAT')
35    READ(1.40.END=50) COMP.START.CODE.PRAT.MACH
40    FORMAT(T2.2I6.I4.I6.I3)
C
C    PATCHER PRODUCTION RATE
C
      PRATE=FLOAT(PRAT)/100000.
C
C    TASK QUANTITY
C
      QTY=FLOAT(COMP-START)*PRATE
C
C    TRANSFORMING DRY VENEER GRADE CODES TO A SINGLE VALUE
C
      COD=DINVEN(GRD(CODE).THK(CODE).LEN(CODE).1)
      PCODE=NPATCH(GRD(CODE).THK(CODE).LEN(CODE).WID(CODE))
      CD=THK(CODE)*100+LEN(CODE)*10+1
C
```

```
C     WRITING THE EVENT THAT WILL REMOVE THE APPROPRIATE QUANTITY FROM
C     DRY INVENTORY
C
        WRITE(2,45) START,QTY,COD,CD,NT(1),NR(1)
C
C     WRITING THE EVENT THAT WILL INCREASE DRY INVENTORY
C
        WRITE(2,45) COMP,QTY,PCODE,CD,NT(1),NR(2)
45      FORMAT(T2,I6,F6.2,2I4,2I2)
        GO TO 35
50      CONTINUE
        CALL CLOSE(1)
        CALL CLOSE(2)
C
C     READING THE TASK ASSIGNED TO THE JOINTER OPERATION
C
        CALL ASSIGN(1,'SY:JOITSK.DAT')
        CALL ASSIGN(2,'SY:JOITK1.DAT')
55      READ(1,60,END=70) COMP,START,CODE,PRAT,MACH
60      FORMAT(T2,2I6,I4,I6,I3)
C
C     JOINTER PRODUCTION RATE
C
        PRATE=FLOAT(PRAT)/100000.
C
C     TASK QUANTITY
C
        QTY=FLOAT(COMP-START)*PRATE
C
C     TRANSFORMING DRY VENEER GRADE CODES TO A SINGLE VALUE
C
        COD=DINVEN(GRD(CODE),THK(CODE),LEN(CODE),1)
        CD=THK(CODE)*100+LEN(CODE)*10+WID(CODE)
C
C     WRITING THE EVENT THAT WILL REMOVE THE APPROPRIATE QUANTITY FROM
C     DRY INVENTORY
C
        WRITE(2,65) START,QTY,CODE,CD,NT(2),NR(1)
        CD=CD-WID(CODE)+1
C
C     WRITING THE EVENT THAT WILL INCREASE DRY INVENTORY
C
        WRITE(2,65) COMP,QTY,COD,CD,NT(2),NR(2)
65      FORMAT(T2,I6,F6.2,2I4,2I2)
        GO TO 55
70      CONTINUE
        CALL CLOSE(1)
        CALL CLOSE(2)
        STOP
        END
```

```
C
C    JULY 8, 1984
C
C    THIS PROGRAM CREATES THE EVENT LIST THAT WILL BE USED IN PROGRAM
C    INVENT TO RECONSTRUCT THE DRY INVENTORY.
C
C
C    AUTHOR:  ERIC SCHILD
C
      PROGRAM EVNTLST
      INTEGER TRNSFR(6),MASTER(2400,6),LINKSR(2400),LINKPR(2400),
     *LSIZE,HEAD,TAIL,ROW,AHEAD,IHEAD,BEHIND,ITAIL,ITEM,NAR
      REAL QTY
      COMMON MASTER,LINKSR,LINKPR,LSIZE,HEAD,TAIL,NAR
C
C    CALLING INITIALIZATION SUBROUTINE
C
      CALL INEVNT
      DO 1000 I=1,4
      GO TO (10,20,30,40) I
C
C    READING MASTER PRODUCTION SCHEDULE
C
10    CALL ASSIGN(1,'SY:PAT1.DAT')
      GO TO 50
C
C    READING EVENT LIST CREATED FROM PATCHER TASK FILE PATTSK.DAT
C
20    CALL ASSIGN(1,'SY:PATTK1.DAT')
      GO TO 50
C
C    READING EVENT LIST CREATED FROM JOINTER TASK FILE JOITSK.DAT
C
30    CALL ASSIGN(1,'SY:JOITK1.DAT')
      GO TO 50
C
C    READING EVENT LIST CREATED FROM DRYER TASK FILE DRYTSK.DAT
C
40    CALL ASSIGN(1,'SY:DRYTK1.DAT')
50    READ(1,55,END=900) TRNSFR(1),QTY,(TRNSFR(M),M=3,6)
55    FORMAT(T2,I6,F6.2,2I4,2I2)
      TRNSFR(2)=NINT(QTY*100.)
C
C    IF NO MORE ROOM IN LINKED LIST, STOP
C
      IF(NAR.GT.0)GO TO 65
      WRITE(5,60)
60    FORMAT(T2,'MASTER STORAGE ARRAY OVERFLOW')
      STOP
65    LSIZE=LSIZE+1
C
C    IF FIRST EVENT, CONTINUE
C
      IF(LSIZE.EQ.1)GO TO 110
      ROW=HEAD
C
C    IF EVENT TIME GREATER THAN OR EQUAL TO TIME OF EVENT IN LINKED
C    LIST, CONTINUE
C
70    IF(TRNSFR(1).GE.MASTER(ROW,1))GO TO 75
C
```

```
C    IF HEAD OF LIST, CONTINUE
C
       IF(ROW.EQ.HEAD)GO TO 100
C
C    NOT HEAD OF LIST
C
C    ASSIGN EVENT AHEAD OF THE CURRENT ROW
C
       AHEAD=LINKSR(BEHIND)
       ROW=NAR
       NAR=LINKSR(ROW)
       IF(NAR.GT.0)LINKPR(NAR)=0
       LINKPR(ROW)=BEHIND
       LINKSR(BEHIND)=ROW
       LINKPR(AHEAD)=ROW
       LINKSR(ROW)=AHEAD
       GO TO 500
C
C    IF EVENT TIME GREATER THAN TIME OF EVENT IN LINKED LIST, CONTINUE
C
75     IF(TRNSFR(1).GT.MASTER(ROW,1))GO TO 80
C
C    IF EVENT INVOLVES REMOVING INVENTORY, CONTINUE
C
       IF(TRNSFR(6).LT.MASTER(ROW,6))GO TO 80
C
C    IF HEAD OF LIST, CONTINUE
C
       IF(ROW.EQ.HEAD)GO TO 100
C
C    ASSIGN EVENT AHEAD THE CURRENT ROW
C
       AHEAD=LINKSR(BEHIND)
       ROW=NAR
       NAR=LINKSR(ROW)
       IF(NAR.GT.0)LINKPR(NAR)=0
       LINKPR(ROW)=BEHIND
       LINKSR(BEHIND)=ROW
       LINKPR(AHEAD)=ROW
       LINKSR(ROW)=AHEAD
       GO TO 500
C
C    ASSIGN EVENT BEHIND THE CURRENT ROW
C
80     BEHIND=ROW
       ROW=LINKSR(BEHIND)
       IF(TAIL.NE.BEHIND)GO TO 70
       ROW=NAR
       NAR=LINKSR(ROW)
       IF(NAR.GT.0)LINKPR(NAR)=0
       ITAIL=TAIL
       LINKPR(ROW)=ITAIL
       LINKSR(ITAIL)=ROW
       LINKSR(ROW)=0
       TAIL=ROW
       GO TO 500
C
C    ASSIGN EVENT AHEAD OF FIRST ITEM IN LIST
C
100    ROW=NAR
```

```
      NAR=LINKSR(ROW)
      IF(NAR.GT.0)LINKPR(NAR)=0
      IHEAD=HEAD
      LINKPR(IHEAD)=ROW
      LINKSR(ROW)=IHEAD
      LINKPR(ROW)=0
      HEAD=ROW
      GO TO 500
C
C   ASSIGN EVENT BEHIND THE LAST ITEM IN LIST
C
110   ROW=NAR
      NAR=LINKSR(ROW)
      IF(NAR.GT.0) LINKPR(NAR)=0
      LINKSR(ROW)=0
      HEAD=ROW
      TAIL=ROW
C
C   TRANSFERRING DATA
C
500   DO 510 ITEM=1,6
      MASTER(ROW,ITEM)=TRNSFR(ITEM)
510   CONTINUE
      GO TO 50
900   CONTINUE
      CALL CLOSE(1)
1000  CONTINUE
      CALL ASSIGN(1,'SY:ENVTLT.DAT')
      ROW=HEAD
C
C   WRITING EVENT LIST
C
1050  IF(ROW.EQ.0)GO TO 1100
      WRITE(1,1075) (MASTER(ROW,M),M=1,6)
1075  FORMAT(I2,2I6,2I4,2I2)
      ROW=LINKSR(ROW)
      GO TO 1050
1100  CONTINUE
      CALL CLOSE(1)
      STOP
      END
```

```
C
C     JULY 8, 1984
C
C     THIS SUBPROGRAM INITIALIZES THE HEAD AND TAIL POINTERS AND
C     PREDECESSOR AND SUCCESSOR POINTERS FOR THE LINKED LIST THAT STORES
C     THE INVENTORY UPDATE EVENTS.
C
C     AUTHOR:  ERIC SCHILD
C
      SUBROUTINE INEVNT
      INTEGER TRNSFR(6),MASTER(2400,6),LINKSR(2400),LINKPR(2400),
     *LSIZE,HEAD,TAIL,ROW,AHEAD,IHEAD,BEHIND,ITAIL,ITEM,NAR
      REAL QTY
      COMMON MASTER,LINKSR,LINKPR,LSIZE,HEAD,TAIL,NAR
      DO 10 ROW=1,2400
      LINKPR(ROW)=0
      LINKSR(ROW)=ROW+1
10    CONTINUE
      LINKSR(2400)=0
      HEAD=0
      TAIL=0
      LSIZE=0
      NAR=1
      RETURN
      END
```

```
C
C     JULY 8,1984
C
C     THIS PROGRAM RECONSTRUCTS DRY INVENTORY TO DETERMINE THE AVERAGE
C     TOTAL INVENTORY AND THE MINIMUM AND MAXIMUM INVENTORY OF EACH DRY
C     VENEER GRADE.
C
C     AUTHOR:  ERIC SCHILD
C
      PROGRAM INVEN
      INTEGER DINVEN(11,4,5,3),INDEX,G,T,L,W,EVTIM,TQTY,CODE,CD,
     *TYPE,NT,RM,THK,LEN,WID,GD(2,11),LENGTH
      REAL BASIS(3,5),STIM,LASTM,FTIM,QTY,TOTVEN,TOTINV,FRC(11),
     *PRATE,ORIQTY,MIN,MAX,RMIN(660),RMAX(660),INVEN(660),TIM,TOTAL
      BYTE GR(4),TH(4),WI(2)
      LOGICAL FLAG
      COMMON CODE,MIN,MAX,INVEN,TOTINV,RMIN,RMAX
      DATA BASIS/.864,.864,.864,1.920,1.824,1.728,2.160,2.052,1.944,
     *2.400,2.280,2.160,2.880,2.736,2.592/
      DATA TOTVEN,TOTINV,INVEN/662*0./
      DATA MAX,RMAX,MIN,RMIN/1322*0./
      FLAG=.FALSE.
C
C     READING DRY VENEER GRADE CODE
C
      CALL ASSIGN(1,'SY:INVCOD.DAT')
      INDEX=1
5     READ(1,10,END=15) G,T,L,W
10    FORMAT(I2,I2,3I1)
      DINVEN(G,T,L,W)=INDEX
      INDEX=INDEX+1
      GO TO 5
15    CONTINUE
      CALL CLOSE(1)
C
C     SETTING UP INITIAL DRY INVENTORY LEVEL
C
      CALL ASSIGN(1,'SY:DRY.DAT')
20    READ(1,25,END=30) G,T,L,W,TQTY,NT
25    FORMAT(I2,I2,3I1,2I5)
      QTY=FLOAT(TQTY)
      CODE=DINVEN(G,T,L,W)
C
C     UPDATING INVENTORY STATISTIC
C
      INVEN(CODE)=INVEN(CODE)+QTY
C
C     UPDATING TOTAL INVENTORY STATISTIC
C
      TOTINV=TOTINV+(QTY*BASIS(W,L))
      CALL STAT
      GO TO 20
30    CONTINUE
      CALL CLOSE(1)
C
      CALL ASSIGN(1,'SY:GRADE.DAT')
      INDEX=1
35    READ(1,40,END=45) (GD(M,INDEX),M=1,2)
40    FORMAT(I2,2I2)
      INDEX=INDEX+1
```

```
      GO TO 35
45    CONTINUE
      CALL CLOSE(1)
C
C   SETTING UP DRYER AND PATCHER DEVELOPMENT FILES
C
      CALL ASSIGN(1,'SY:ENVTLT.DAT')
      OPEN(UNIT=2,TYPE='OLD',ACCESS='DIRECT',NAME='SY:DRYDEV.DAT',
     *RECORDSIZE=15,READONLY)
      OPEN(UNIT=3,TYPE='OLD',ACCESS='DIRECT',NAME='SY:PATDEV.DAT',
     *RECORDSIZE=15,READONLY)
C
C   READING EVENT FROM EVENT FILE
C
50    READ(1,55,END=95) EVTIM,TQTY,CODE,CD,TYPE,RM
55    FORMAT(T2,2I6,2I4,2I2)
      THK=IFIX(FLOAT(CD)/100.)
      LEN=IFIX(FLOAT(CD-THK*100)/10.)
      WID=CD-(LEN*10+THK*100)
      QTY=FLOAT(TQTY)/100.
      TIM=FLOAT(EVTIM)/1440.
C
      IF(FLAG)GO TO 60
C
C   SETTING UP TIME MECHANISM
C
      STIM=TIM
      LASTM=TIM
      FLAG=.TRUE.
C
C   UPDATING TOTAL INVENTORY STATISTIC
C
60    TOTVEN=TOTVEN+TOTINV*(TIM-LASTM)
C
C   ADVANCING TIME
C
      LASTM=TIM
C
C   IF EVENT INVOLVES UPDATING INVENTORY, CONTINUE
C
      IF(RM.EQ.0)GO TO 85
C
C   EVENT INVOLVES UPDATING INVENTORY
C
C   IF EVENT WAS NOT GENERATED FROM TASK ASSIGNED TO PATCHER OPERATION,
C   CONTINUE
C
      IF(TYPE.GT.2)GO TO 70
C
C   EVENT WAS GENERATED FROM PATCHER OPERATION
C
      READ(3'CODE) (FRC(GD(1,M)),M=1,8),PRATE
      DO 65 I=1,8
      IF(FRC(GD(1,I)).EQ.0)GO TO 65
      CODE=DINVEN(GD(1,I),THK,LEN,1)
      ORIQTY=QTY*FRC(GD(1,I))
C
C   UPDATING INVENTORY STATISTIC
C
      INVEN(CODE)=INVEN(CODE)+ORIQTY
```

```
      CALL STAT
65    CONTINUE
      GO TO 50
C
C    IF EVENT WAS NOT GENERATED FROM TASK ASSIGNED TO JOINTER OPERATION.
C    CONTINUE
C
70    IF(TYPE.GT.3)GO TO 75
C
C    EVENT WAS GENERATED FROM JOINTER OPERATION
C
C    UPDATING INVENTORY STATISTICS
C
      INVEN(CODE)=INVEN(CODE)+QTY
      CALL STAT
      GO TO 50
C
C    EVENT WAS GENERATED FORM DRYER OPERATION
C
75    READ(2'CODE) (FRC(GD(2,M)),M=1,8),PRATE,(GR(M),M=1,4),
     *(TH(M),M=1,4),LENGTH,(WI(M),M=1,2)
      DO 80 I=1,8
      IF(FRC(GD(2,I)).EQ.0)GO TO 80
      CODE=DINVEN(GD(2,I),THK,LEN,WID)
      ORIQTY=QTY*FRC(GD(2,I))
C
C    UPDATING INVENTORY STATISTICS
C
      INVEN(CODE)=INVEN(CODE)+ORIQTY
C
C    UPDATING TOTAL INVENTORY STATISTIC
C
      TOTINV=TOTINV+(ORIQTY*BASIS(WID,LEN))
      CALL STAT
80    CONTINUE
      GO TO 50
C
C    IF EVENT WAS NOT GENERATED FROM MASTER PRODUCTION SCHEDULE, CONTINUE
C
85    IF(TYPE.GT.1)GO TO 90
C
C    EVENT WAS GENERATED FROM MASTER PRODUCTION SCHEDULE
C
C    UPDATING INVENTORY STATISTIC
C
      INVEN(CODE)=INVEN(CODE)-QTY
C
C    UPDATING TOTAL INVENTORY STATISTIC
C
      TOTINV=TOTINV-(QTY*BASIS(WID,LEN))
      CALL STAT
      GO TO 50
C
C    EVENT WAS GENERATED FROM EITHER JOINTER OR PATCHER OPERATION
C
C    UPDATING INVENTORY STATISTIC
C
90    INVEN(CODE)=INVEN(CODE)-QTY
      CALL STAT
      GO TO 50
```

```
95    CONTINUE
      CALL CLOSE(1)
      CALL CLOSE(2)
      CALL CLOSE(3)
C
C   CALCULATING TOTAL INVENTORY
C
      ETIM=TIM
      TOTAL=TOTVEN/(ETIM-STIM)
      CALL ASSIGN(1,'SY:INVST.DAT')
      WRITE(1,100) MIN,MAX,TOTAL
100   FORMAT(T2,3F10.2)
      DO 110 I=1,660
      IF(INVEN(I).EQ.0)GO TO 110
      WRITE(1,105) I,RMIN(I),RMAX(I)
105   FORMAT(T2,I4,2F10.2)
110   CONTINUE
      CALL CLOSE(1)
      STOP
      END
```

```
C
C    JULY 8, 1984
C
C    THIS SUBPROGRAM EVALUATES THE INVENTORY STATISTICS AFTER EACH
C    UPDATE IN PROGRAM INVEN TO DETERMINE THE MAXIMUM AND MINIMUM
C    LEVELS.
C
C    AUTHOR:  ERIC SCHILD
C
      SUBROUTINE STAT
      INTEGER CODE
      REAL MAX,MIN,TOTINV,INVEN(660),RMAX(660),RMIN(660)
      COMMON CODE,MIN,MAX,INVEN,TOTINV,RMIN,RMAX
      IF(TOTINV.LE.MAX)GO TO 10
      MAX=TOTINV
10    IF(TOTINV.GE.MIN)GO TO 20
      MIN=TOTINV
20    IF(INVEN(CODE).LE.RMAX(CODE))GO TO 30
      RMAX(CODE)=INVEN(CODE)
30    IF(INVEN(CODE).GE.RMIN(CODE))RETURN
      RMIN(CODE)=INVEN(CODE)
      RETURN
      END
```