# Design of Interactive Systems for Real-Time Dobby Control

W.D. Hoskins and J.A. Hoskins

ABSTRACT

The design of control systems for a microcomputer-controlled dobby loom is examined with respect to establishing the critical performance features and display options required by the designer/weaver. Descriptions of the interactive architecture are given in terms of event-driven diagrams that permit the straightforward translation of the principles to any computer system. Idealized display forms are proposed which provide the weaver with a variety of possible classes of information about the current session, ranging from production information to structure, color and block detail.

## Introduction

In considering the problem of designing a microcomputer assisted loom for a professional to use in the serious production of woven fabrics, it is necessary to evaluate the *complete environment used* or *potentially used* by that person [**1**, **2**]. Almost seemingly extraneous parts of the situation can have an immediate negative effect on either productivity or creativity, and features of minor physical or psychological annoyance at the beginning of a time period can become, over an extended period of time, either a real physical discomfort or of apparent unbearable clumsiness.

The traditional interest of computer science with the perfection of the user interface has therefore to be extended to encompass:

1. The physical positioning of the weaver at the loom.
2. The probable sequence of actions implied by the production process.
3. The performance limitations of the weaver.
4. The performance limitations of the loom.
5. The performance limitations of the dobby control mechanism.
6. The performance limitations of the computer(s) and displays.
7. Error recovery and corrections introduced by the weaver.
8. Interactive Information.

ARS TEXTRINA 5 (1986), pp. 33-49

The dobby control problem will be discussed with respect to the control and solution of these eight points.

## The physical equipment and its arrangement

The system used for the development of the control system was :

### Weaving equipment:

1. An Ahrens and Violette 48" 16-shaft dobby loom of Jack type.
2. A flying shuttle.
3. An underslung beater.
4. An automatic tension system.

### Electronics (prototype system):

1. A parallel Ahrens and Violette COMPU-DOBBY system.
2. An APPLE II+ microcomputer with two 5.25" disk drives.
3. An AXLON 320k ram disk system providing two virtual drives.
4. Two CRT green screen monitors.
5. An 80 column card.
6. A parallel interface card for communication with the dobby unit.
7. A parallel interface dot matrix printer and GRAPPLER+ interface card.

### Electronics (present system):

1. A 512k MACINTOSH microcomputer with external drive.
2. A 20 megabyte hard disk.
3. An IMAGEWRITER serial printer.
4. A serial Ahrens and Violette COMPU-DOBBY unit.
5. User-programmable battery-powered cartridges.

The second of the two computer systems used reflects the continuing development of the equipment involved in this area and also the refinement of the interface towards transparency.

The general appearance of the equipment is given in Figure 1; however, a schematic illustration of their physical arrangement for utilitarian purposes appears in Figure 2.

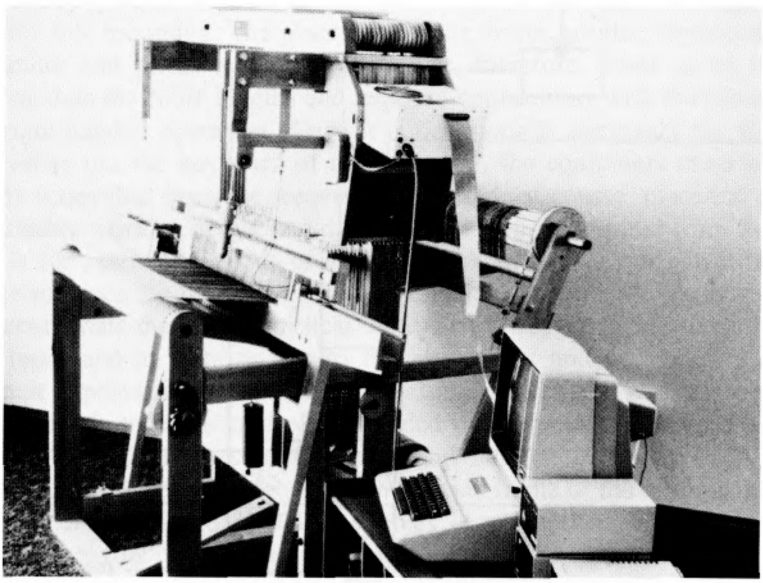


Figure 1

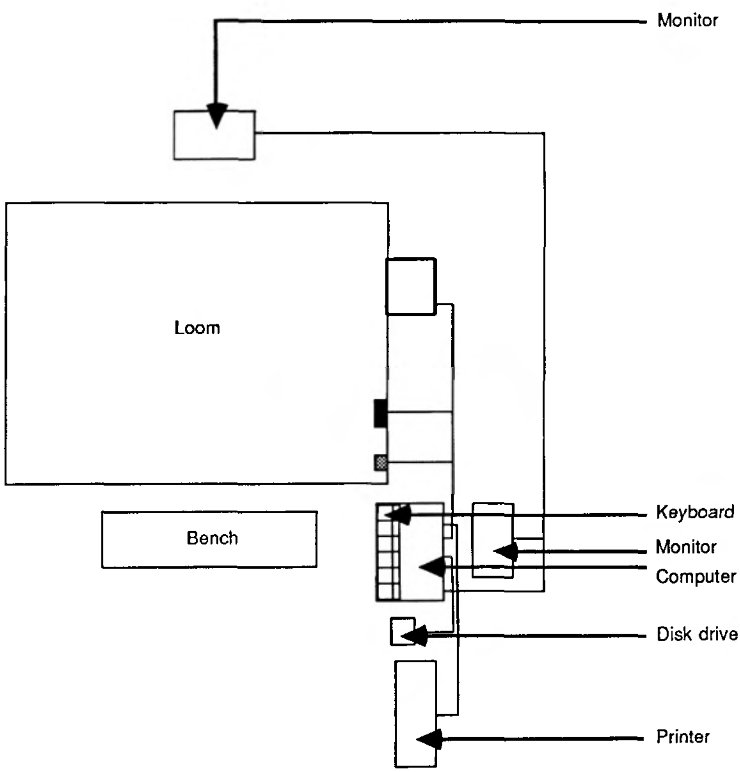

Monitor
Loom
Bench
Keyboard
Monitor
Computer
Disk drive
Printer

Figure 2

Figure 2 shows the arrangement of the parts of the computer and the relationship of the weaver's position at the bench, relative to the loom. The second monitor was positioned so that, during normal weaving conditions, a slight movement of the eyes would make the screen of the monitor visible through the overhead part of the loom and above the shafts, even when the shafts were raised. Of necessity, the organization of the equipment was for right-handed operation since the structure of the dobby (illustrated schematically in Figure 3) requires a right-handed side mounting. The placement of the drives, printer, keyboard, computer and primary CRT monitor was, therefore, made so as to accomodate the cable lengths and required connections with the loom, for right-handed operation. Since it is occasionally necessary for the weaver to use the keyboard of the computer, the equipment must be easily accessible from the weaver's bench and, of course, placed at a reasonable working height and distance. Use of the computer from the bench for extended periods of time is not comfortable, nor can it be made so, since the bench is a loom bench and ergonomically designed to accomodate the active physical motion of a weaver facing towards the loom and at right angles to the electronics and computer. An attempt to provide the weaver with comfortable access to a reasonable spectrum of software control is afforded by the buttons mounted on one of the side frames of the loom and marked in Figure 3. An additional hardware item mounted on the side frame of the loom is the optical sensor (Figure 3) which enables the computer to determine whether weaving is taking place.

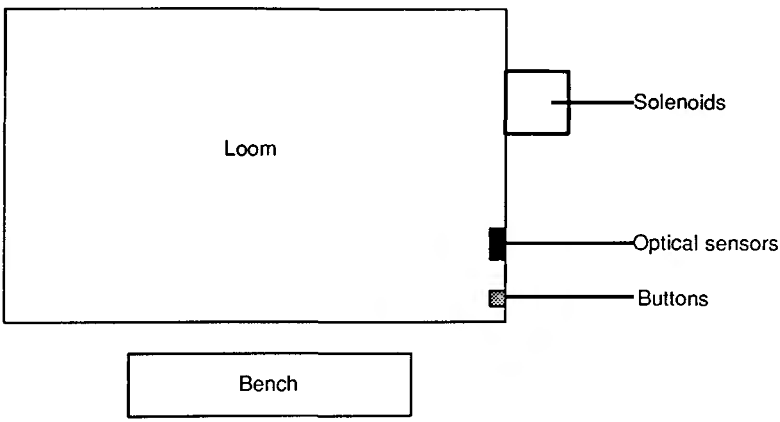



Figure 3

## Examining the computer-loom dialogue

Superficially, the relationship between the parts of the system appears to be as in Figure 4. There is a computer and a loom and, presumably, two way communication (indicated by the arrows on the lines). It must be a two way communication, of course, since the machine must set the state of the solenoids which control which combination of shafts will be raised when the treadle is depressed, but must not reset the solenoids for the next pick until the present pick has been inserted and beaten into the fell.

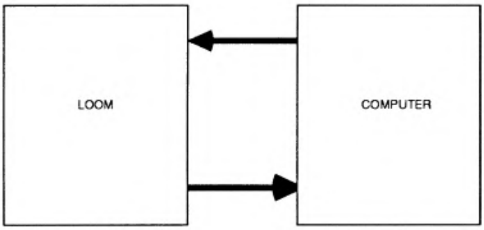



Figure 4

In actuality, the discussion of the previous paragraph skipped over the third element of the trialogue, namely the weaver. The depression of the treadle is an element of the three way dialogue of critical importance to both the computer and the loom and a more correct description of the relationships is given in Figure 5.

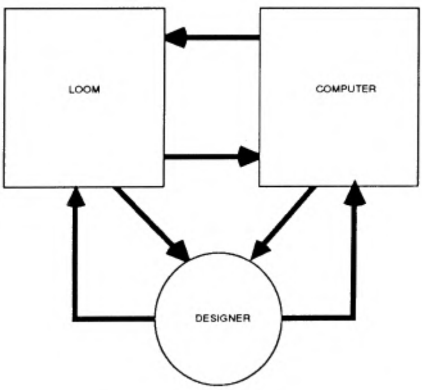



Figure 5

Again, it is relevant to note that the paths indicated in the figure are all two-way and are, explicitly:

Designer to Loom:

1. Insertion of a pick.
2. Depression of the shedding treadle.
3. Possible advancement of the cloth.
4. Possible finger manipulation of other yarns.

Loom to Designer:

1. The visual appearance of the cloth as it is woven.
2. The sound of the shedding mechanism and dobby solenoid unit.
3. The tension and feel of the cloth and warp sensed through the action of the beater and insertion of the weft pick.
4. The weight and feel of the shafts through the treadle and shedding mechanism.

Designer to Computer:

1. A physical modification to pick sequencing, by using the buttons on the loom frame.
2. A more overt modification performed by using the keyboard for input to the microcomputer.
3. Pressing a loom button.

Computer to designer:

1. Graphical and textual displays of various types.
2. Auditory warnings optionally set to indicate possible important aesthetic or functional breakpoints in the work in progress, e.g. end of border, beginning of design.

Loom to Computer:

1. Present state of the solenoids.
2. Arm change of state.
3. Which button was pressed.

Computer to Loom:

1. Set solenoids to ...

The pattern of physical interactions is given by Figure 6, and is necessary in determining the subsequent critical event path.

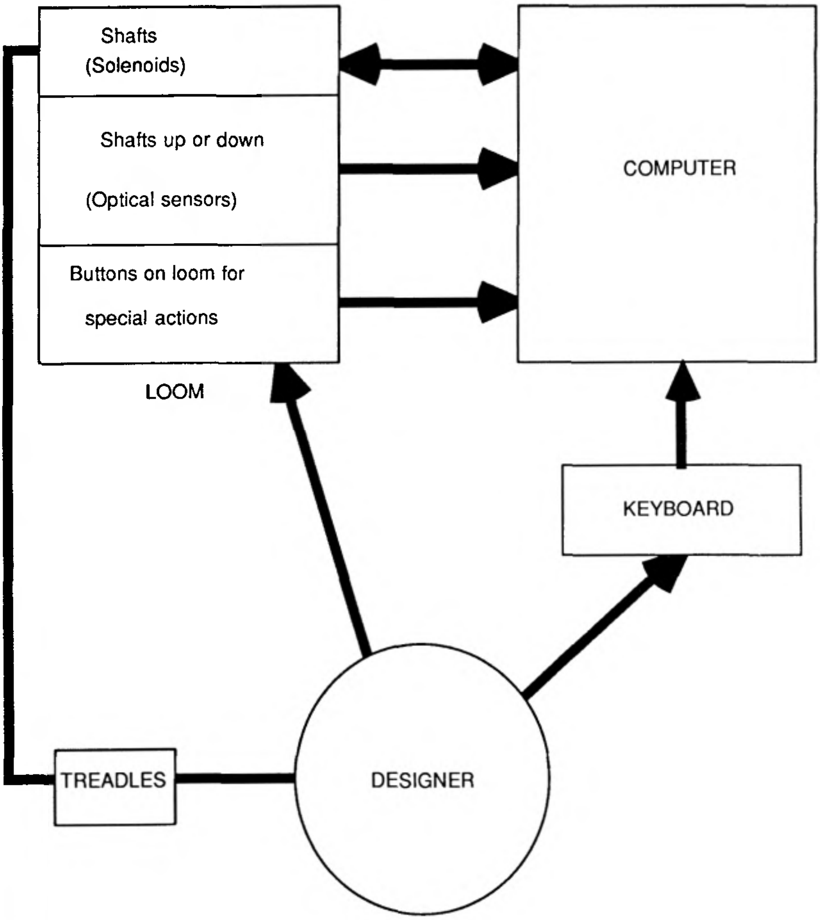
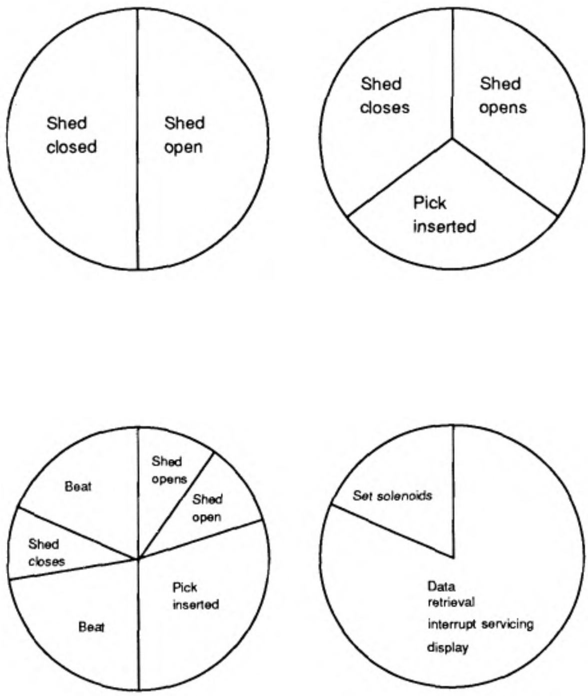



Figure 6

## The weaving event cycle

At its simplest level the loom is a device for repeatedly holding the shed either open or closed. Therefore, any pattern of actions involving the loom must take this into account. The periodic nature of this sequence is best expressed by a circle (see Figure 7) so that the complete cycle is seen to involve a structured sequence of actions taking place in a given time period. These diagrams do not necessarily indicate a relative allocation of time to the contained action or state, so that, for instance, the second diagram in Figure 7 more accurately portrays the shedding sequence, since opening or closing a shed takes place in a finite period of time. Clearly, from this second diagram, pick insertion can take place in the time that the shed opening is wide enough for the pick to be inserted. Including more details of the weaving process leads to the third diagram where now the action of the beater in placing the pick is related to the state of the shed (open or closed) and this event, which is generated by the weaver, is bound as an ordered event in time to the state of the shed. The fourth diagram in Figure 7 shows that, to the computer, the weaving cyle is relatively simple. The solenoids are set in one specific time period and then the machine is free to handle all the other tasks presently necessary, such as retrieving data for subsequent picks, updating graphical displays, coping with an interrupt outside of the usual weaving sequence, or updating production



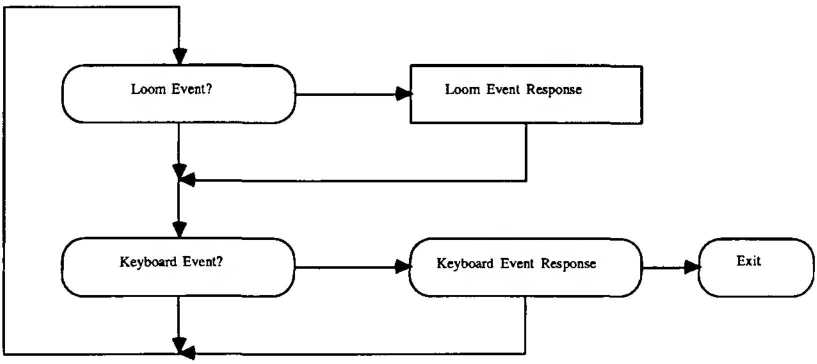

Weaving Cycles
Figure 7

statistics. It is the reconciliation of these four descriptions of time and their realisation as a control system which creates the complete dobby environment.

Ensuing sections will deal successively with the detailed elaboration of this event hierarchy.

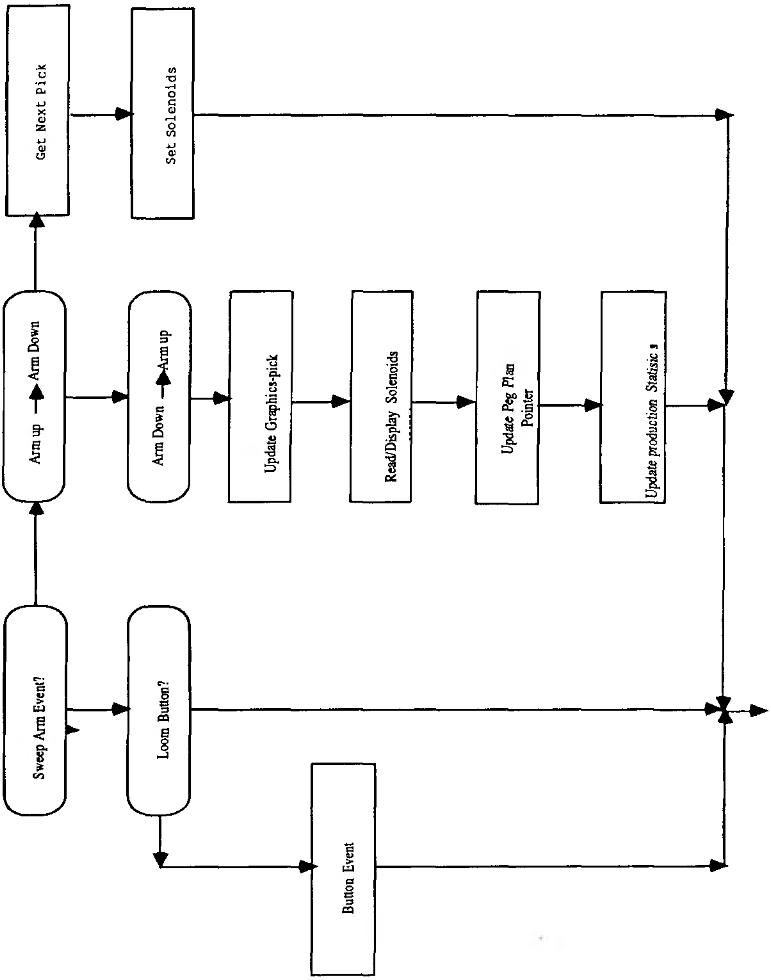
### Event sequencing

The basic description of the event relationships is given in Figure 8, where it is apparent that the main software loop is concerned with checking, on a continuous basis, whether a loom event or a keyboard event has been generated. Both of these are, of course, asynchronous actions, occurring at various times and not at a fixed separation in time. The main loop illustrated though, is timing critical, in that it must execute faster than events are generated. Otherwise, the weaver will be faced with the situation of over-running the weaving speed of the loom with no "weave ahead buffer". Exit from the software weaving state is naturally a keyboard action, so as not to be selected accidentally.




Dobby Weaving Environment
Figure 8

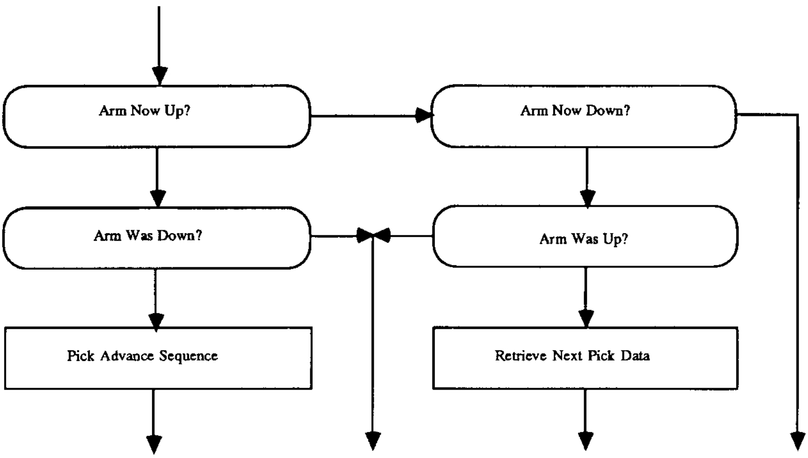
The two actions described in Figure 8, can be further broken down, as in Figure 9, where the possible events generated at the loom and communicated to the microcomputer are related. Critical in this case is the path from "sweep arm event" down through "loom button?" and the paths containing "get next pick" and "read/display solenoids". Both of these latter paths are time critical in that they actually lie in the weaving loop. The placing of "get next pick" in a path where there is slack time saves time in the "read/display solenoids" path and achieves the practical aim of providing next pick data, for graphical display to the weaver. Other significant features of this interface are the generation




Loom Event
Figure 9

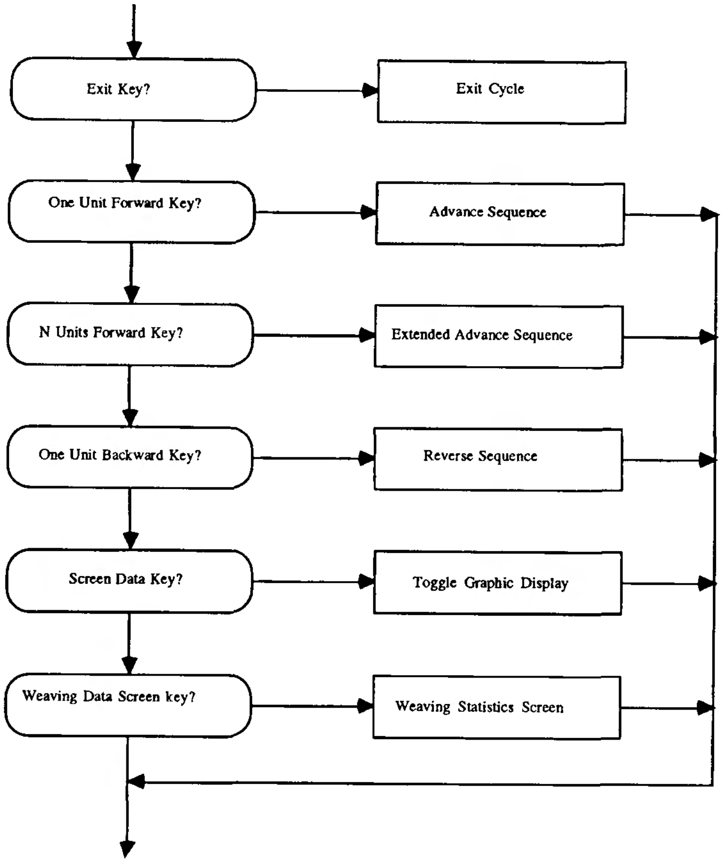
of a peg plan pointer (a feature of the grahical display of the woven fabric) and the ability to accumulate production statistics. These could be, number of metres woven, number of picks/nano-century or number of repeats.

A complete examination of that part of Figure 9 dealing with the arm state is given in Figure 10, where the precise relationship of the arm state to the possible paths is given. Note that, in both figures, it is assumed that an arm state change event has been detected.




Sweep Arm Event
Figure 10

Keyboard events are simpler and of more obvious utility, so that, being able to manually advance the weaving position in the peg plan, either a single unit or multiple units, in either direction, is clearly necessary. A method of exiting from the weaving mode is also required. If the graphic display mode forweaving is not shared with a screen displaying production statistics then a key to toggle between the two forms of data is needed (Figure 11).




Keyboard Event Hierarchy
Figure 11

## Keyboard action sequences

The keystroke events discussed in the previous section require, in all cases, an orderly updating and preservation of the current data for the weaving loop state. Typically, to execute the exit sequence, the solenoids in the dobby unit must be turned off (no power state), and pertinent data, such as the row index in the pegplan, the color of the current pick, the current block index, and present tabby index, must be stored. In addition, the production statistics must be updated and closed and the graphics screens set appropriately (see Figure 12).

Exit Cycle

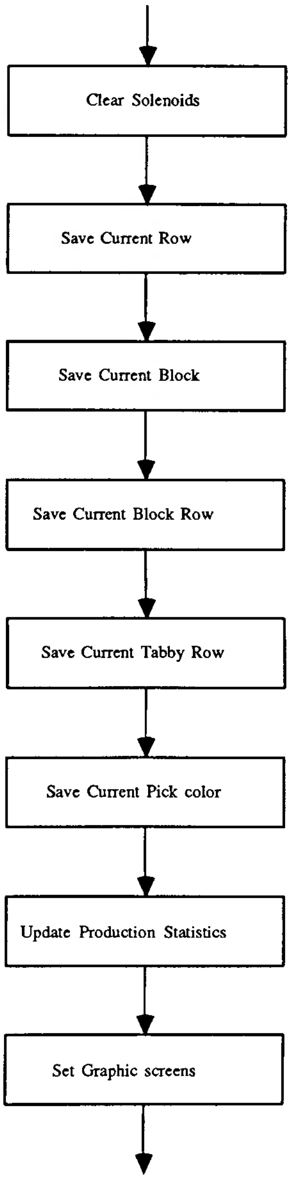
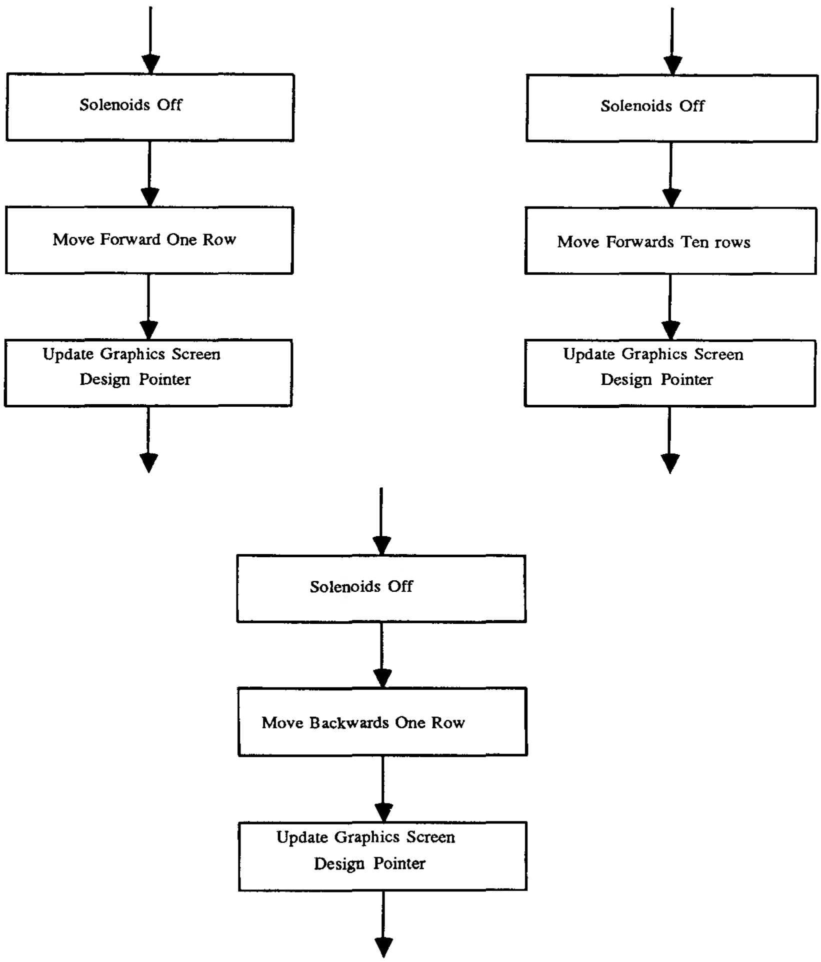



Figure 12

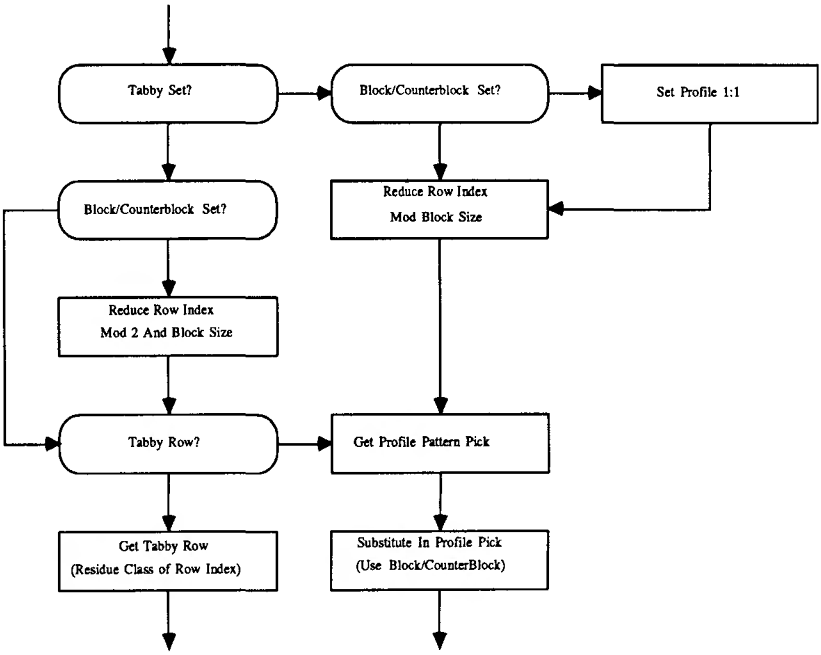
Similar actions are necessary in the manual movement of the pointer to the current lag selected in the peg plan, and are given in Figure 13.




Advance / Reverse Sequences
Figure 13

## Data retrieval

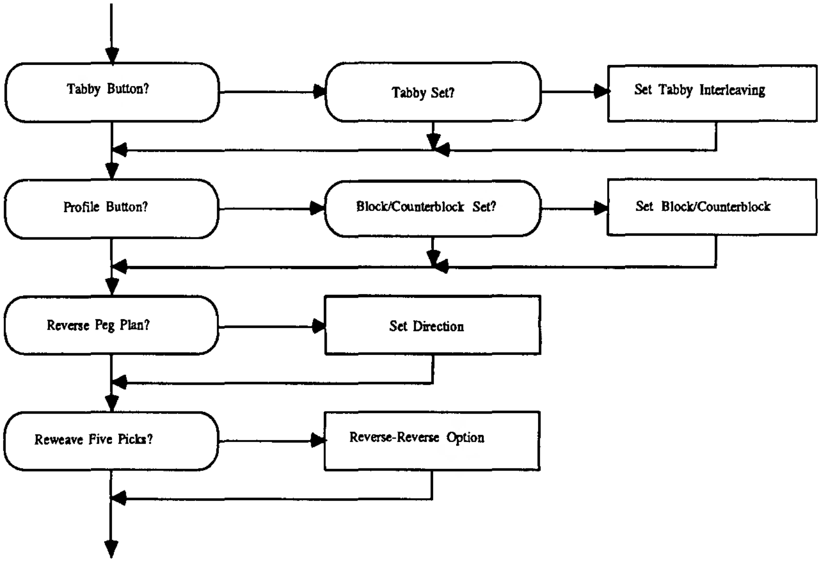
The manipulation and retrieval of the peg plan data is described in Figure 14, where it is plain that, if the peg plan is to be considered a peg plan profile description, then the data must be retrieved in the weaving loop indirectly, through a previously selected block/counter-block weave (e.g. 5 shaft satin/sateen). Additionally, if the block weave has tabby interleaved between pattern picks (e.g. Summer and Winter) then the current pick to be woven and, hence, also the solenoid settings, must be determined with respect to both the block row index and the tabby index. More complex features, such as automatic variation of the ratio of picks in face cloth and back cloth could also be incorporated at this time.




Pick Retrieval
Figure 14

## Button events

The four possibly useful actions available to the weaver seated at the loom are given in Figure 15, where it is apparent that the buttons do




Button Events
Figure 15

no more than manipulate the flags controlling the block and tabby substitution described in the previous paragraph, as well as controlling the increment/decrement index for a specified, or absolute, number of rows.

## REFERENCES


[1] J.A. Hoskins and M.W. King, *An interactive database for woven textile design,* Textile Institute Annual Conference, "Computers in the World of Textiles", Hong Kong, September, 1984.

[2] J.A. Hoskins, W.D. Hoskins and M.W. King, *A microcomputer-controlled dobby loom for textile prototyping,* International Symposium on Fiber Science and Technology, Japan, August, 1985.



Department of Computer Science,
University of Manitoba,
Winnipeg, Canada, R3T 2N2