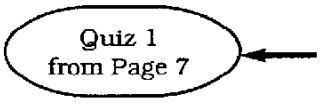
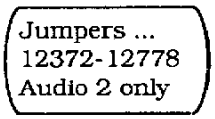
Example Flowchart Symbols

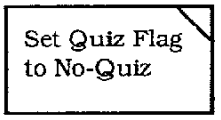
The symbols below (or similar ones) could be used in a flow chart to aid the design process and to document the interactive aspects of an Audio / Video presentation. The flowchart is a graphic representation of both the order of information presentation and of how the interactive control program responds to viewer inputs. The flowchart is the interactive "story board", which should be used to guide subsequent scripting and programming. Consistency, clarity, and completeness are more important than the symbols used. When in doubt, use text descriptions of your intentions.



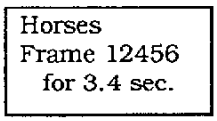
A terminal symbol, indicating the starting or ending point of a flowchart, or a continuation of the flowchart from another page.



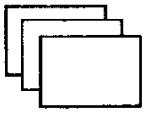
A motion sequence, with or without sound. Possibly, a still image with narration or music.



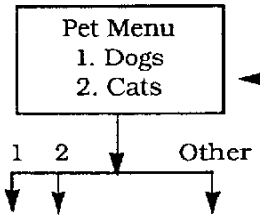
Special instructions, scoring, or any computations the control program may be required to do. Flags may be set, values recorded, etc.



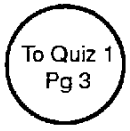
A still frame. Include how long to wait at the still before continuing to the next block in the flowchart.



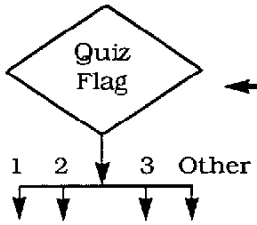
A series of stills. Specify how the user proceeds forward through the stills. Is it by time steps only? Can the user back up in the sequence? What happens when the user goes "past" the end still, or tries to back up "past" the beginning still?



A still frame Menu. All possible viewer inputs are accounted for, and a timeout, if any, is specified. The branch paths, and the input values that cause the path to be taken, are clearly shown.



Indicate a continuation of the flow chart on another page. Include the page number.



A decision point, indicating the value or condition to be tested, the branching that results, and the values that caused the branching.

Sample Level II Pogramming Code

Some Necessary Definitions

The Examples in this Appendix were written to help explain how to create Level II programs. To simplify these examples, they were written in symbolic assembler form. This means that a program called a symbolic assembler will be necessary to convert any of the examples to actual code that a Pioneer videodisc player can run. The input to a symbolic assembler is called the "source code file" and is in a syntax that is easy for a programmer to understand; the output of an assembler is called the "object code file" and contains the codes the videodisc player understands.

Symbolic assemblers offer various methods for a programmer to control the process of converting the symbolic source code file to the final object code file. The following is a description of the assembler control syntax used in the four examples of this Appendix. The assembler you use may require different syntax to accomplish the same operations.

```
RSEQ1$ 20                ; Set the value of symbol RSEQ1 to 20
RSEQ2$ 22                ; Set the value of symbol RSEQ2 to 22

$N 23                    ; $N sets the assembler's internal register pointer to register 23

$R 2300 2000 1200 1000   ; $R loads the data into registers 23, 22, 21, and 20 respectively

RSTS1 EQU 24             ; Set the value of symbol RSTS1 to 24
RSTS2 EQU 26             ; Set the value of symbol RSTS2 to 26

$N 27                    ; $N sets the assembler's internal register pointer to register 27

$R 3057 3050 3010 3001   ; $R loads the data into registers 27, 26, 25, and 24 respectively

$ADDR 0                  ; Set the assembler's internal program counter to 0, all code
                         following this statement will be entered into successive locations
                         in memory.
```

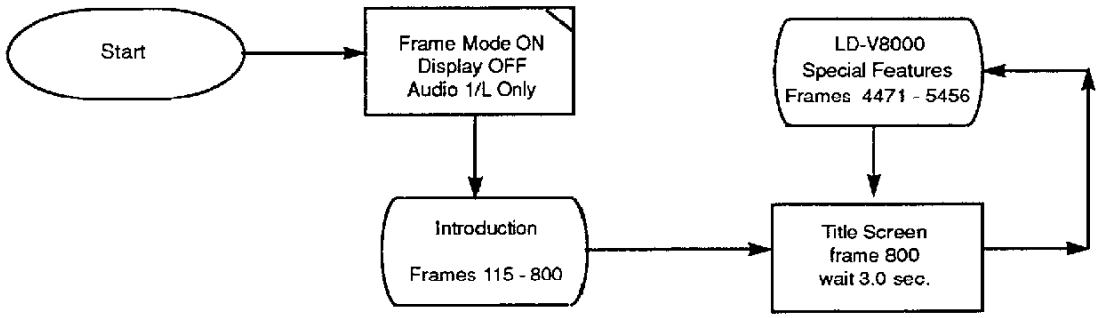
Sample Level II Program Code

Level II Example #1 - Flow Chart

For use with LD-V8000 Demo Disc — CAV Side

A Repeating Video Segment, preceded by an Introduction.

This flowchart details the introductory sequence and the motion segment to be looped.



Level II Example #1 - Program Code

Continuously Repeating Video Segment

The Level II source code shown below uses a mixture of command names and mnemonics. The intent is to show an operational program in an educational way. Some Level II compilers may require a slightly different syntax. With hand entry of the program, the Program Address of each label (TITLE below) must be noted and that address value substituted where necessary.

```
                $ADDR 0     ; START THIS PROGRAM AT PROGRAM ADDRESS 0000
                SFM
        0       DISPLAY     ; TURN OFF DISPLAY
                ANF         ; TURN AUDIO 1 AND 2 ON
        115     SEARCH      ; CUE BEGINNING OF MOTION
        800     AUTOSTOP    ; PLAY THE INTRODUCTION

TITLE:
        800     SEARCH      ; CUE BEGINNING OF MOTION
        30      WAIT        ; SHOW TITLE FRAME FOR 3 SEC
        4471    SEARCH      ; CUE BEGINNING OF MOTION
        5456    AUTOSTOP    ; SHOW LOOP MOTION SEGMENT
        TITLE   BRANCH      ; LOOP BACK
```

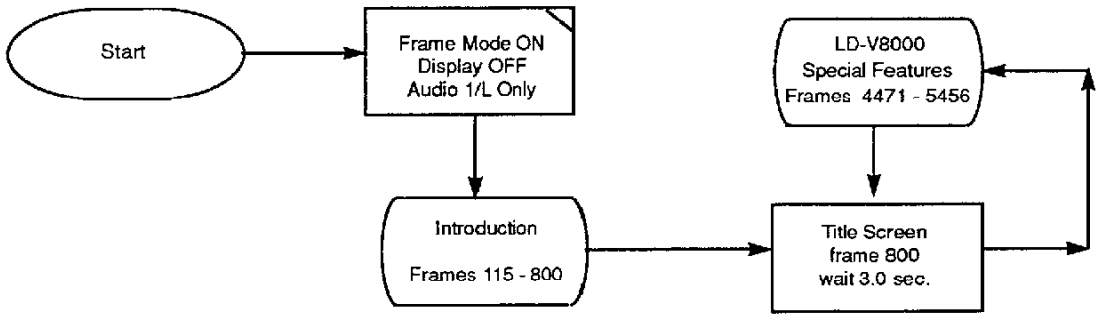
Sample Level II Program Code

Level II Example #1 - Flow Chart

For use with LD-V8000 Demo Disc — CAV Side

A Repeating Video Segment, preceded by an Introduction.

This flowchart details the introductory sequence and the motion segment to be looped.



Level II Example #1 - Program Code

Continuously Repeating Video Segment

The Level II source code shown below uses a mixture of command names and mnemonics. The intent is to show an operational program in an educational way. Some Level II compilers may require a slightly different syntax. With hand entry of the program, the Program Address of each label (TITLE below) must be noted and that address value substituted where necessary.

```
                $ADDR 0     ; START THIS PROGRAM AT PROGRAM ADDRESS 0000
                SFM
        0       DISPLAY     ; TURN OFF DISPLAY
                ANF         ; TURN AUDIO 1 AND 2 ON
        115     SEARCH      ; CUE BEGINNING OF MOTION
        800     AUTOSTOP    ; PLAY THE INTRODUCTION

TITLE:
        800     SEARCH      ; CUE BEGINNING OF MOTION
        30      WAIT        ; SHOW TITLE FRAME FOR 3 SEC
        4471    SEARCH      ; CUE BEGINNING OF MOTION
        5456    AUTOSTOP    ; SHOW LOOP MOTION SEGMENT
        TITLE   BRANCH      ; LOOP BACK
```

Sample Level II Program Code (cont.)

Level II Example #2 - Program Code (CONT.)

```
ATTRACT:
            100             SC          ; ATTRACT LOOP 100 - 700

PRESSING:                               ; WAIT FOR NO KEY PRESS
            BIN             GET         ; GET NUMBER OF LAST KEY PRESSED
            255             COMPARE; IF KEY NUMBER IS NOT 255 KEY WAS PRESSED
            PRESSING        BR
                            PLAY
            SPIN            BR          ; WAIT FOR NO-BUTTON
            PRESSING        BR          ; AND START PLAYING

SPIN:                                   ; LOOP HERE AND CHECK FOR END OF ATTRACT LOOP
            0               RECALL
                            STORE       ; GET CURRENT FRAME #
            690             COMPARE; SEE IF ALMOST DONE
                                        ; WE NEED A LITTLE WARNING BEFORE THE REAL
MOTION END

DONE:                                   ; ATTRACT LOOP NEAR END SO
            700             AS          ; FINISH LAST 10 FRAMES
            ATTRACT         BR
            DONE            BR          ; NOT YET TO FRAME 690, SEE IF ANY BUTTON PRESSED
            BIN             GET         ; CHECK IF USER PRESSED A KEY
            255             COMPARE
            MMENU           BR          ; ERROR, SHOW MENU -- CAN'T REALLY GET HERE
            SPIN            BR          ; NO BUTTON PRESS
            MMENU           BR          ; BUTTON PRESS, SHOW MENU

;----------THE END ----------------------------------------------
```