TABLE OF CONTENTS

1 INTRODUCTION AND OVERVIEW

1.1 TITLES

1.1.1 SCENE

Vegmatic titler

1.1.2 ACTION

Star Wars type titles give one sentence purpose of MIDAS and the videotape.

1.1.3 REALIZATION

(Exact title goes here.)

1.1.4 TIMING

30 sec

1.2 INTRODUCTION

1.2.1 SCENE

Narrator is shown against BUGS room working background.

1.2.2 ACTION

The narrator states the purpose of MIDAS and gives some caveats about its use, background, and an overview of what is to be shown following. Emphasis in this narration is on illustrating handshaking and asynchronous operation in digital systems. A disclaimer is made saying that we are showing only a subset of the tecnical facilities available in MIDAS, and of the possible uses for which it is intended. The viewer is advised that the display quality is much better online than in the videotape.

1.2.3 REALIZATION

(Shot of narrator in front of BUGS)

NARR. The purpose of this presentation is to demonstrate some of the capabilities of MIDAS, which stands for Microprocessor Instructional Display and Animation System. MIDAS is intended for instructional use, to illustrate the operation of a complete microcomputer system using real-time interactive graphics techniques. MIDAS was specifically designed to illustrate concepts of asynchronous control and the low level control interaction between devices, often called "handshaking." An integral part of the system is a discrete simulation of the CPU and system peripherals which include a Bus Control Unit, Priority Interrupt Control Unit, DMA and Floppy Disk Controller, and a Console and Keyboard Interface. The microprocessor on which this simulated system is based, is the Intel 8080.

(Cut to narr. in front of MIDAS user at VG)

As we will see later, MIDAS includes many facilities that make it useful as an interactive teaching tool. These include the ability to single step through a program, replay states of the simulation, and freeze it at any time. The user is given the capability of panning over the display, zooming in on levels of detail, and controlling the speed of the simulation interactively during operation, using a joystick and analog dials.

(Shot of MIDAS user typing commands)

A command language is also provided which allows the user to load and save programs or the entire state of the

system on disk, and modify the contents of registers, flags, lines and memory within the simulated system.

(Cut back to narr.)

What follows is a brief demonstration of MIDAS illustrating the 8080s interrupt handling. Since this presentation is only a preview of MIDAS, we are able to show only a limited amount of the system's capabilities. In the future, with more extensive presentations, we hope to be able to give a more complete synopsis of MIDAS' use as a teaching tool. Also, keep in mind that the quality of the display suffers due to the limited resolution of the video system. Online no such degradation takes place, and all text and lines of the display are highly readable.

1.2.4 TIMING

30 sec

1.3 MOTIVATION

1.3.1 SCENE

Three different methods of conventional instruction are shown: a professor in the classroom with a blackboard; a blueprint in a laboratory setting; and a student studying INTEL 8080 documentation.

1.3.2 ACTION

The camera cuts to a scene of a student in a classroom trying to understand a lecture on the Intel 8080, with a diagram of the simulated system on the blackboard. From there, a cut is made to the same student puzzling over Intel 8080 documentation.

1.3.3 REALIZATION

1.3.4 TIMING

10-15 sec for each of the three cuts

45 sec total

1.4 COMPARISON OF MIDAS WITH CONVENTIONAL TEACHING METHODS

1.4.1 SCENE

Top level of MIDAS display on VG scope. MIDAS is in Normal Mode at highest speed.

1.4.2 ACTION

The camera shows a tight shot of the VG scope face showing only the MIDAS basic display. The narrator briefly describes advantageous similarities between MIDAS and the methods shown in 1.3 He then describes the advantages of MIDAS type instruction over old methods.

The narrator then goes through a catalog of animation effects with the border of the basic display (containing menus, etc.) in the picture. The pan-zoom capability is described while MIDAS pans back and forth over the top level of the display. As MIDAS then zooms in and out of the display twice, the narrator describes the various levels of detail being shown on the screen. MIDAS then zooms back to the top level of the display, and, while the narrator describes the real time animation technique and emphasizes user control in real time, the camera pans back, encompassing the student of 1.3 operating the system.

1.4.3 REALIZATION

1.4.4 TIMING

Pan/zoom 10 sec
Levels of detail 20 sec
Animation effects 30 sec
User control 10 sec

70 sec total

1.5 OVERVIEW OF THE BASIC DISPLAY

1.5.1 SCENE

VG scope view of top level of MIDAS in Command Mode (no action).

1.5.2 ACTION

On the top level of detail of the MIDAS display, the narrator describes each, emphasizing that all such effects are accomplished by MIDAS, and not the camera. device of the simulation in one sentence, while MIDAS pans over the appropriate box. The narrator then describes the animation conventions concerning the lines, registers and flags, and the bus transfers, while zooming in on examples of each. Finally, the narrator describes the liberties the system takes with the actual 8080 chips and peripherals' clocking, bus terminations, and addressing.

The camera then cuts to titles showing the three scenarios to follow, as the narrator briefly outlines them.

1.5.3 REALIZATION

(Cut to top level MIDAS display.)

NARR. This is the top level of MIDAS' basic display. Each device in the simulation, is represented by a labelled box on the diagram. Interconnecting the devices are single bit control lines, such as these lines from the Console to the BCU...

(Zoom in on BECON/CONHLD and flash them)

...which are brightened when high and dim when low. The wider paths represent busses.

(Zoom in on data bus from CPU to STAT; start up in NORMAL mode, slowest speed.)

Here we see the eight bit data bus from the CPU. Note that bus transfers are indicated by the hex equivalent for the data moving across the bus to its destination.

(Back to top level display.)

An important capability of the system is the ability to zoom in on the display and have finer details of the diagram appear as we move closer. Notice more detailed representations appear as we come closer to focus on a particular unit. Note that all zoom and pan effects shown here are accomplished by MIDAS and not the camera.

(Zoom in on CPU PC reg very slowly from top level.)

Here we zoom in on the 8080's register and flag complement. The program counter is shown with the hex equivalents of its contents. When register or flag contents change, blinking is added to highlight the activity.

(Cut back to narrator)

1.5.4 TIMING

30-45 sec

2 SYSTEM DEMONSTRATION

2.1 BASIC MACHINE CYCLE

2.1.1 SCENE

Top level of MIDAS display on VG scope is shown. The first demo program is loaded and execution proceeds in Normal Mode at the slowest speed.

2.1.2 ACTION

From the top level of detail of the display, MIDAS switches to standard view and slowly zooms in to the level of the programmer accessable registers in the CPU. The narrator describes the state organization of the machine cycle of the Intel 8080 briefly, and the instruction --> machine cycle --> clock state hierarchy of the instruction cycle.

Then, at the slowest speed, we follow the execution of a single non-memory referencing instruction (CMA), while the narrator describes the action using pointer and possibly replay for highlighting. The narration anticipates events (such as register transfers) in the display.

2.1.3 REALIZATION

2.1.4 TIMING

approx 120 sec

2.2 INTERRUPT HANDLING

2.2.1 SCENE

Standard view on MIDAS display. The system is running in Normal Mode at slow speed. The CPU is in wait state with interrupts enabled, and the stack pointer set to point to a non-zero memory location for saving the return address.

2.2.2 ACTION

MIDAS pans to the console interrupt line and follows the interrupt sequence from interrupt initiation at the console, back to the standard view. The narrator briefly describes the interrupt structure of the 8080 and the RST instruction. The narration follows the vectored interrupt from the PICU and the interrupt acknowledge sequence until the interrupt handler is executing.

2.2.3 REALIZATION

NARR. We will now load a simple program into the 8080s memory from BUGS disk. The program sets up the 8080 stack pointer to point at location x'200' in memory, sets the PICU mask register to accept an interrupt from the console, and enables the 8080 to recognize interrupts. We start the interrupt handling sequence following execution of the HALT instruction.

(Cut to MIDAS std view)

Note the setting of the WAIT flag on the 8080, indicating that it is in wait state. When the sequence starts, an interrupt will be generated from the console in response to the user's hitting the simulated console interrupt key. The interrupt will be passed from the PICU to the 8080. This will take the CPU out of wait state and cause a RESTART instruction to be taken from the PICU. The restart instruction will cause a call to an interrupt handling routine at one of eight possilbe low memory locations, selected by three bits within the op-code.

Since it is a subroutine call, the return address will be saved on the stack.

(SSTEP)

Here the interrupt key on the console has been hit, raising the interrupt line to the PICU.

(SSTEP)

As the interrupt line to the CPU is raised, and the 8080 responds by coming out of wait state. The CPU has entered a cycle called "interrupt acknowledge after halt."

(SSTEP -- T1)

Here we see status for that cycle being sent out on the data bus to the status decoder...

(SSTEP -- T2)

...which in state T2 raises the interrupt acknowledge line to the PICU. It responds by sending a jammed instruction to the CPU.

(SSTEP -- T3)

In state T3 of the cycle, the restart instruction is read into the instruction register of the CPU...

(SSTEP -- T4)

...where it is internally decoded in state T4. Also, the W register is loaded with zero, the address of the interrupt handler in this case.

(SSTEP -- T5)

In state T5 the stack pointer is decremented in anticipation of the next cycle of the instruction, a stack write.

(SSTEP -- T1 STACK WRITE)

In state T1 of the stack write cycle, the address in the stack pointer register is written out on the address bus, while status is sent out on the data bus.

(SSTEP -- T2)

Now the high order byte of the program counter is written out to the stack and the stack pointer is decremented for another stack write cycle.

(Run normal mode -- moderate speed)

Next the low order byte of the program counter is written on to the stack. Finally, the restart is completed as the address of the interrupt routine is written out on the address buffer from the WZ register. During the next instruction fetch, the program counter will be updated from the WZ register, completing the branch to the interrupt handler.

(Cut back to narr.)

2.2.4 TIMING

approx 120 sec

2.3 DMA HANDLING

2.3.1 SCENE

2.3.2 ACTION

2.3.3 REALIZATION

2.3.4 TIMING

approx 120-180 sec

2.4 DEBUGGER FACILITY

2.4.1 SCENE

Top level of MIDAS display. System is in Command Mode, with a demo program loaded and several instructions already executed.

2.4.2 ACTION

The display on the VG scope changes from the top level MIDAS display, to the debugger display. The narrator describes the purpose of the debugger as an aid to developing the simulation, as well as a natural alternative way of displaying digital information for debugging and instruction. The narration mentions the possiblities of use in teaching hardware maintenance.

2.4.3 REALIZATION

2.4.4 TIMING

30 sec

3 CONCLUSION

3.1 RETROSPECTIVE

3.1.1 SCENE

Final cut back to top level MIDAS basic display. The system is in Real Time mode.

3.1.2 ACTION

Over real time display of MIDAS, the narrator summarizes the purpose of the system. The narration re-emphasizes the illustration of asynchronous operation and handshaking. It emphasizes the reduction of learning and teaching time when MIDAS is used as a teaching aid in combination with classroom lectures. It describes the extensibility of this type of system to other processors than the Intel 8080, or even other components than microprocessors. The final message is that this type of dynamics is very useful for illustrating complex technical subjects with multiple levels of detail. The narration also mentions the merits of user controlled, real time interactive techniques that MIDAS uses in instruction.

3.1.3 REALIZATION

NARR. What you have just seen is a brief demonstration of MIDAS. The capabilities and potential uses of the system go beyond what we have been able to show you in this short presentation. MIDAS is a well suited for illustrating the complex activities of asynchronous operations and handshaking protocol. The real value of the system can be seen in its use as a teaching aid in a computer architecture course or in a course meant to familiarize engineers or potential customers with a specific computer system. Its potential for making educational films is, of course, another use.

The techniques used in MIDAS are general ones. This type of display could be extended to simulations of other digital systems. We have found that systems such as MIDAS can greatly reduce teaching and learning time. They constitute a more easily understood method of illustrating dynamic processes than conventional static teaching methods.

3.1.4 TIMING

60 sec

3.2 CREDITS

3.2.1 SCENE

Vegmatic titler

3.2.2 REALIZATION

(Credits go here.)

3.2.3 TIMING

?? sec