Pulse width modulation control in DLP™ projectors

Abstract: *Digital Light Processing™ (DLP) display systems are based upon the Digital Micromirror Device™ (DMD™), a spatial light modulator developed by Texas Instruments. Light intensities from the DMD are produced by pulse width modulation (PWM) of the mirrors over the operating refresh time. Generally stated, PWM is accomplished by successively loading video data bit planes onto the DMD and resetting the device.*

This article describes the design methodology utilized to direct the DMD in this manner. A number of logical units have been developed to translate the PWM bit sequence, which is designed with product and image requirements in mind, into the detailed memory and reset control functions of the device. The challenge was that all units developed were custom designed and implemented to satisfy the unique and complex requirements of PWM for the DMD while maintaining a simple, flexible user interface.

The units associated with the design of PWM control for DLP products are described in the article and are summarized as follows:

- *DMD divided reset controller (DRC). This hardware module has been designed to control memory and reset operations of the device simultaneously. The DRC supports a set of reset and memory instructions that comprise the DMD divided reset hardware language (DRHL).*
- *Sequence PROM. A collection of DRHL programs is stored in a sequence PROM. Each program specifies the sequence of DMD control for a given input video frame rate range. These programs can be accessed randomly using address vectors.*
- *DMD control language compiler (DCLC). This software application compiles an assembly-level set of sequence programs into machine-level DRHL downloadable onto the sequence PROM.*
- *DMD PWM instruction generator. This software application takes a high-level text-based description of the PWM sequence, as well as electronics and system parameters associated with the product, creating the assembly-level set of sequence programs for input into the DCLC.*

The Digital Micromirror Device (DMD) uses pulse width modulation (PWM) to achieve various levels of light intensity on each of the pixels in the DMD array. By turning a pixel mirror on and off in response to data bits at a rate faster than the human eye can perceive, the pixel appears to have an intermediate intensity proportional to the fraction of time the mirror is on.

When representing each pixel of a video frame digitally using a multi-bit word, the bits of the same significance from all pixel words are called a bit plane. If the elapsed time the mirrors are left in the state directed by each bit plane is proportional to the relative bit plane "weighting," the pixels create the desired picture.

In practice, a particular bit plane may be split and repeated many times per frame for efficiency and appearance purposes. The DMD is typically divided into reset blocks, updated individually. In sequential color systems, the PWM must be coordinated with the color filter cycle. Finally, systems are usually required to operate over a range of frame rates. These conditions make the control of PWM very complex and have led to the development of several hardware and software modules. These modules aid the design process while preserving flexibility of this control through all operating modes of DMD video products.

Overview

This article describes three of the major modules developed to meet requirements associated with the design and implementation of PWM control in Digital Light Processing (DLP) systems. *Figure 1* shows an overview of these modules as they inte-

Don Doherty and Greg Hewlett

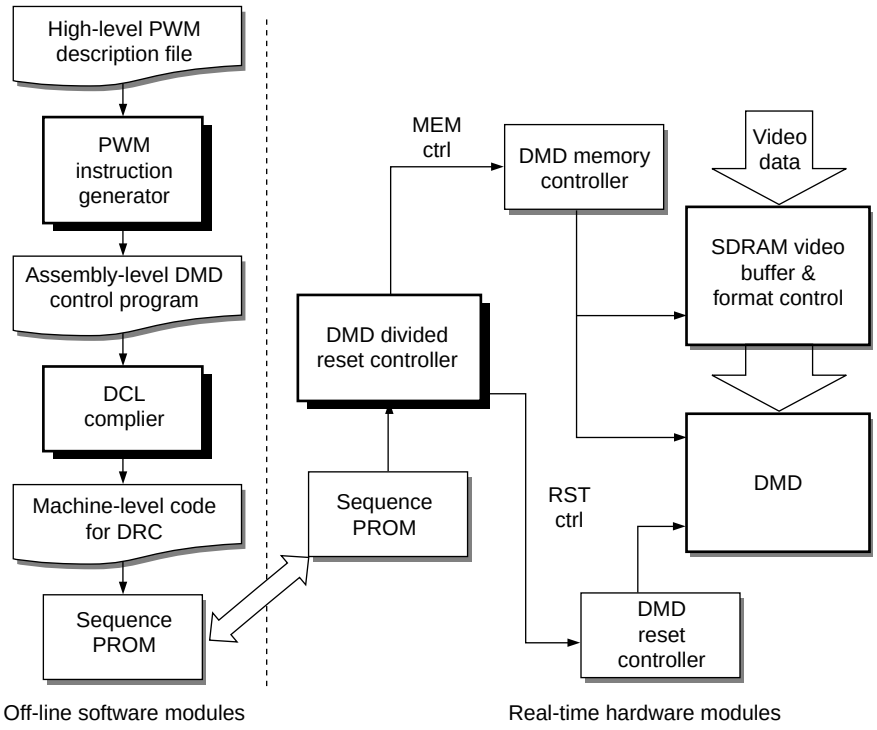

Figure 1. PWM sequence software and hardware flow.

grate with off-line software and real-time hardware for PWM control.

Video data is placed into a SDRAM buffer to be loaded onto the DMD, one bit plane and reset block at a time. A PWM sequence program is a set of instructions directing loading and resetting of the DMD over one video frame. The divided reset controller (DRC) controls this process by executing sequence programs stored in the sequence programmable read-only memory (PROM). The DRC is composed of two independent units that handle both video data memory load and reset control of the DMD.

The code in the sequence PROM is generated off-line by the DMD control language compiler (DCLC). The DCLC takes an assembly-level program for each required video input frequency range and generates machine level binary data loaded onto the sequence PROM. This data contains a set of reset and memory instructions for each frequency sub-range supported by the product. For example, one sequence set of instructions might support all video-input rates from 59.6 - 60.6 Hz, while another would support 60.9 - 61.6 Hz.

The assembly-level code as generated by the DMD PWM instruction generator (DMD-PIG) provides a high-level PWM sequence description. This input file is designed with product specific goals such as image quality, video input, and color filter specification.

Divided reset controller

The DMD divided reset controller has been developed to translate detailed PWM sequence machine instructions into precise timing signals for other system circuits. One set of timing signals goes to the DMD memory controller while another set goes to the DMD reset controller. *Figure 2* shows a block diagram of the DRC.

One of the functions the memory controller can perform is specified bit plane block transfer of the current video frame into the section of the DMD corresponding to the selected block. The length of time required to perform such a transfer is one of the parameters that determine the limits on sequence design. This block transfer time is derived from the hardware characteristics of the memory controller and DMD. The DRC initiates a block data transfer by sending the memory controller a block address and bit plane ID along with a signal to start the transfer. The memory controller then executes the transfer with no further signaling from the DRC. Since the transfer time is fixed and known, the PWM sequence design can

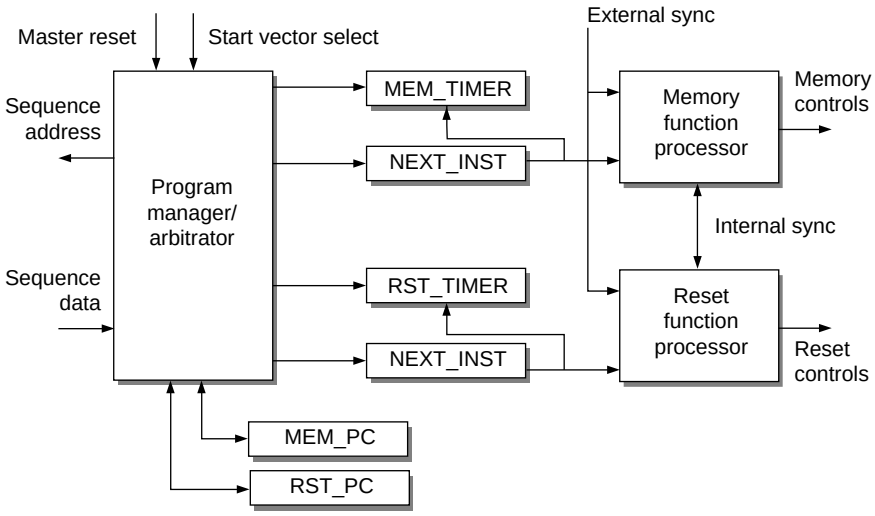

Figure 2. Divided reset controller block diagram.

preclude subsequent transfer requests from happening before the memory controller is ready.

Another memory controller function clears (sets to zero) DMD mirror cells. In this case, the DRC simply tells the memory controller to start the clear function. The time required to complete the clear operation is another example of sequence design parameters. The clear function is used whenever all DMD mirrors are to be turned off at the end of a display frame or when the color of incident light must be changed in a sequential color system.

The reset controller generates high voltage waveforms that signal DMD mirrors to change state. The standard reset waveform is programmed into the reset controller as well as other waveforms that are used for displaying bits with very short duration. The reset controller can apply a reset waveform to all DMD mirrors simultaneously or to an individual reset block that corresponds to a block of data cells. The DRC tells the reset controller what type of waveform to generate and provides a start signal. In addition, the DRC either gives the reset controller a block address identifying an individual reset block or a code to reset the entire DMD.

The DRC executes two separate machine language programs simultaneously, one for memory functions and the other for reset functions. Since both programs are stored in the same sequence PROM, the DRC maintains separate program counters (MEM_PC and RST_PC illustrated in Figure 2) and pre-fetches instructions as needed for each program. The DRC also provides separate counter registers (MEM_TIMER and RST_TIMER) for memory and reset functions that control the time delay between functions. These counters are loaded with data provided in each memory / reset instruction and then count down to determine when the next function of the same type executes. In this way, the DRC controls the timing of memory and reset functions to the resolution of its basic clock period. The program manager / arbitrator regulates the fetching of sequence instructions from the PROM, including all branching and access conflict resolution.

The sequence PROM contains a section of start vectors (address 0) that hold start addresses of memory and reset programs for one or more sequence programs. The DRC uses an externally supplied address to select the start vector and begin execution of a PWM sequence. The use of start vectors allows multiple PWM sequence programs of variable length to be stored in the sequence PROM. By using this start procedure periodically (such as the beginning of each video frame), an external controller can select from a suite of PWM sequences that are individually optimized for various conditions like frame rate or scene content.

DMD control language compiler

The instructions in the sequence PROM that control DMD memory and reset functions contain time delays between functions in DRC clock cycles. The calculations required to generate delay counts are performed by the DMD control language compiler. The DCLC is a program developed for PC and UNIX platforms that translates a list of instructions (similar

to assembly code) into DRC usable machine code. In generating machine code and included delay counts, the DCLC must ensure memory and reset programs stay on the same time base and that truncation errors do not accumulate as the PWM sequence programs execute.

The five types of instructions that comprise DCLC input code are start vectors, compiler directives, branching and synchronization, memory functions and reset functions. The DCLC also provides for label addressing. Start vectors only occur sequentially at the beginning (PROM address "0"). Their arguments are labels that equate to the PROM addresses of memory and reset code sections.

Compiler directives define certain hardware parameters such as the DRC clock rate and delay counter size, but do not generate machine instructions. Other compiler directives manipulate status masks inserted into subsequent machine instructions used for external synchronization.

Branching and synchronization instructions generate machine code that can interrupt program execution by, for example, waiting on an external start signal or fetching a new start vector. Some of these instructions establish a reference time from which all following times are calculated.

The principal parts of DCLC code contain separate sections of memory instructions and reset instructions. These functional instructions cause the generation of machine code instructions on a one-to-one basis. Each memory and reset instruction consists of a mnemonic, identifying the function and arguments (if required) such as bit plane ID and time. The time argument, in microseconds, can be specified as an absolute, based on a previously defined reference or relative to that of the preceding instruction. In either case, the time argument identifies when that function is supposed to commence. Since the machine instruction contains a delay count to the next function execution, the DCLC must look ahead to the next function time before the delay can be known.

The DCLC resolves these function times into integral DRC clock cycles so that the delay counts can be calculated. This process is performed on a cumulative basis so that each instruction begins on the nearest clock cycle count from the defined reference time. This approach ensures that cumulative delay counts of memory and reset machine code are never more than one DRC clock period apart. The critical timing relationship between memory and reset operations is thereby maintained.

The DCLC allows PWM sequences to be defined by a readable language, and can conveniently combine multiple sequences into a single module to support a particular product in a variety of applications.

DMD PWM instruction generator

The DMD PWM instruction generator, a software tool that runs on a UNIX platform, takes a high-level text-based description of the PWM sequence and generates assembly-level files for DCLC compiling. The DMD-PIG is useful not only in that it greatly eases the task of writing PWM instruction code, but also generates image quality metrics used in sequence design. Thus, a sequence designer can optimize a particular bit ordering with direct feedback on image quality, without having to generate a sequence PROM for subjective testing on a projector.

The DMD-PIG comprises approximately five thousand lines of code, which automatically perform algorithmic functions. These include resolving reset conflicts among DMD blocks, "phasing" of DMD blocks for phased reset operation, Fourier transform-based computation of image quality metrics, and assurance of a linear light transfer function output. The following sections describe the input and output of the DMD-PIG.

Input

Each set of PWM bit sequences is specified by a description file, which is read by the DMD-PIG. An example of a sequence description file is shown in *Figure 3*.

The *define* command is used to specify electronics timing information, mirror response data, etc. The *colordef* command is used to specify data about each color, including the ASCII name given to the color. The luminance efficiency value is used to compute PWM efficiency data that directly relates to system brightness. The *bitweight* command allows independent specification of non-binary weightings to be assigned to a given bit. In the example of Figure 3, bit 4 of blue will receive a weighting of 16.1/255.1 rather than the default weighting of 16.0/255.0. The sequence command is used to specify the actual bit ordering of color sequences. This command also specifies the begin time with respect to sequence start and duration of each sequence. The illustrated bit ordering and duration times shown here are simplified for example purposes.

Besides the input data specified in the PWM sequence description file, a set of command-line input parameters is provided to activate various fea-

```
. example partial file for sequence
. (single chip DLP system)
.
. include #xga_data.dmd

define MIRROR_TRANSIT_TIME_RST        3.2
define RESET_SEQUENCE_TIME                    1.304
define DATA_HOLD_TIME                         16.0
define NUMBER_RESET_GROUPS                    15
. (many more parameters deleted here)

.           sbpo   name     luminance efficiency
colordef    0      Green    0.7125784
colordef    8      Red      0.1274958
colordef    16     Blue     0.0743477

bitweight Blue 4 16.1

sequence Red      7 5 6 4 7 3 5 2 7 1 6 7   G0   B200.0     T5000.0
sequence Green    7 5 6 4 7 3 5 2 7 1 6 7   G0   B5400.0    T5000.0
sequence Blue     7 5 6 4 7 3 5 2 7 1 6 7   G0   B10600.0   T5000.0
```

Figure 3. Sequence description file.

tures of the DMD-PIG. These features include a diagnostic image output, various levels of timing data output and an option to generate code for earlier versions of the DRC.

Process

Using parameters specified in the sequence description file, the DMD-PIG computes the portion of the video frame period that must be allocated to each bit necessary to keep the PWM proportional to the bit weights. Detailed DMD and general design rules are then applied to allocate each bit's time into segments throughout the frame. These bit segment times yield absolute on and off periods during a frame for every reset block of each bit plane. Finally, the DMD-PIG uses these on and off times to generate memory and reset sequence program code.

Output

The DMD-PIG generates two types of output, UNIX *stdout* (to the screen) and assembly-level PWM code file(s). The screen output is information found to be helpful in the design of optimized sequences for given products. For example, there are many trade-offs in deciding whether to "split" bits and/or "re-order" them. The DMD-PIG gives statistics that show what directions might be taken by the designer to produce the optimal sequence.

Besides the screen output, the final output from the DMD-PIG is the actual assembly-level code to be compiled by the DCLC. There are two sets of instructions: memory (loads, clears, etc) and reset (reset block, reset-release, DMD bias-on, etc). *Figure 4* shows extracts from each type of code.

Conclusion

Because of the unique and complex requirements, PWM for DLP systems has presented the challenge of developing custom software and hardware so that product requirements are met. Three logical units were described which have supported PWM func-

Reset code sample

```
rstb    10417.15702,8    ;   rst   blk    8
rstb    10426.32369,9    ;   rst   blk    9
rstb    10435.49036,10   ;   rst   blk    10
rstb    10441.69778,1    ;   rst   blk    1
rstb    10444.65703,11   ;   rst   blk    11
rstb    10450.86445,2    ;   rst   blk    2
```

Memory code sample

```
ldb     10405.03610,4,14   ;   ld bit 4,   blk 14
ldb     10414.20277,4,15   ;   ld bit 4,   blk 15
ldb     10423.36944,4,16   ;   ld bit 4,   blk 16
ldb     10432.53611,6,1    ;   ld bit 6,    blk 1
ldb     10441.70278,6,2    ;   ld bit 6,    blk 2
ldb     10450.86945,6,3    ;   ld bit 6,    blk 3
```

Figure 4. Samples of PWM sequence code.

tionality, while maintaining the flexibility needed to adapt to differing DLP products.

Some of the major production and implementation requirements are met with the modules described in this article are shown in *Table I.*

References

1. L. Hornbeck, "Digital Light Processing" (invited paper), SID EuroDisplay 96, The 16th International Display Research Conference, Birmingham, England, October 1996.
2. G. Sextro, T. Ballew, and J. Iwai, "High-Definition Projection System Using DMD Display Technology," *SID 95 Digest*, pp. 70-73, 1995.
3. V. Markandey and R. Gove, "Digital Display Systems Based on the Digital Micromirror Device", SMPTE 136th Technical Conference and World Media Expo, October 1994.
4. Hewlett and W. Werner, "Electronic Cinema Projection with the Texas Instruments Digital Micromirror Device," 137th SMPTE Technical Conference, September 1995.
5. D. Doherty and G. Hewlett, "Phased Reset Timing for Improved Digital Micromirror Device (DMD) Brightness," SID 1998 International Symposium, Anaheim, Calif., May 1998.

Trademarks

Requirement	Module	Result
Loading of independent bit planes, often repeating a given bit	DRC	Picture quality improvement
Accommodation of independent reset blocks on the DMD with independent reset and memory control (support phased reset operation)	DRC	Brightness, picture quality improvement
Ability to support multiple frequency ranges through the use of multiple sequence programs	DCLC; Sequence PROM	Versatility in video input rate
Support of high-level sequence design	DMD-PIG	Reduction of production cycle time and R&D cycle time
Metric output of off-line tools	DMD-PIG	Improvement in R&D efficiency and effectiveness

Table I. Production and implementation requirements.

Don Doherty

Donald Doherty is a systems engineer for the Digital Imaging business where he designs system and component architectures and performs video quality analysis and related improvements for DMD-based products. Since joining Digital Imaging in 1992, he has led an ongoing effort to understand and advance the application of Digital Light Processing to the human visual system.

Employed by TI since 1976, Don worked on a variety of Defense Systems and Electronics Group projects including 13 years in Electro-Optics. He designed electronics and software for several infrared searching and tracking applications and participated in the successful flight test of the Silent Attack Warning System (SAWS).

Don was elected Senior Member, Technical Staff in 1995 and holds 14 U.S. patents.

He received a master of science in electrical engineering in 1979 from the University of Texas at Arlington and a bachelor of science in electrical engineering from Texas Tech University in 1975 where he was inducted into the Tau Beta Pi and Eta Kappa Nu societies.

Outside TI, Don and his wife enjoy traveling and bicycling.

Greg Hewlett

Gregory Hewlett has been a systems engineer for Digital Imaging (DI) since 1993. His current assignment includes development of the pulse-width modulation timing subsystem for Digital Light Processing systems. Other assignments at DI have included application of human visual system models to improving image quality of DI video systems, and research in video compression and video signal processing.

Before joining DI, Greg worked in the TI Speech and Image Understanding Laboratory (SIUL). While in SIUL, he developed DSP applications for speech synthesis and analysis.

He received a bachelor of science in electrical and computer engineering from Rice University in 1990, with a double major in art / art history. In 1992 he received a master of science from MIT, where he served a research assistantship developing Cheops, a real-time video coding system at the MIT Media Lab.

Greg is a member of the Society of Motion Picture and Television Engineers and the IEEE. From 1995-1997 he served as TI's representative to the VESA Plug and Display Committee, where he helped draft the Plug and Display Standard. He holds three U.S. patents in digital video technology.

Currently, Greg is working half time for TI, while attending Westminster Theological Seminary in Philadelphia, where he is working toward a master of divinity degree. He enjoys reading, music and spending time with his wife, Christine. ❑