Pong Video Game Using the MSP430

Giancarlo Javier Lujan and Murugavel Raju *MSP430*

ABSTRACT

This application report and its associated software demonstrate the ability of the MSP430 to generate NTSC video signals in real time using the Timer_A Module. A pong video gaming system is implemented using the generated NTSC video signals and a standard television is used to display the game. Figure-1 shows a simulated frame of the Pong video game during play.

Figure 1. Simulated Frame of the Pong Video Game During Play

Introduction

The Pong video game was one of the first video games to be released in the early 70's. Creating an image for video games requires a dedicated video chip to read the image data from the graphics memory. In this application the MSP430F1121A microcontroller (MCU) with its internal RAM configured as the graphics memory functions as the video chip to generate monochrome video signal in real time.

Theory of NTSC Video

A single frame of the National Television Systems Committee (NTSC) standard video is comprised of 525-scan lines, which are interlaced into two vertical frames (even and odd) with a refresh rate of 60-Hz per vertical frame. On a conventional television screen an electron beam builds the required image by scanning through the screen. The horizontal scan lines on the screen are synchronized by horizontal sync pulses and the vertical frames are synchronized by vertical sync pulses. Additionally equalization pulses are provided to stabilize the electron beam as it sweeps across the screen. Each scan line has a 63.5-μs period. The first 4-μs of that period is a sync pulse, which informs the television that a new line is coming, the next 8-μs is a sync pulse used for the television to position the electron beam, and remaining 52-μs is used to write image data onto the screen.

MSP430 Pong Implementation

A minimum of three analog voltage levels are required by the video signal to generate a monochrome image on a television: 0-Volts for sync level, 0.3-Volts for black level, and 1-Volt for white level. The generation of these voltage levels is done by a 2-bit digital-to-analog converter (DAC), which is a simple resistor divider network based on R6 and R7 driven by the general-purpose I/O pins P1.0 and P1.1. The video signal termination impedance of 75-Ohms serves as the load resistor for this resistor divider based DAC. Figure-2 is the schematic of the MSP430 Pong video game board.

This complete Pong video game system is comprised of a MSP430F1121A MCU, a TPS77001 low dropout regulator (LDO), an 8-MHz crystal oscillator, two digital controller connectors, an audio and video connector, and a JTAG connector to provide debug and programming capability.

The LDO regulates the 9-Volt battery power supply to 3.6-Volts and R1 and R2 provide the feedback to stabilize the LDO output at 3.6-Volts. This regulated 3.6-Volt supplies the VCC of the MSP430F1121A and the external digital controllers. The 8-MHz ceramic resonator with integrated load capacitors is connected to XIN and XOUT of the MCU and provides the clock for the system. The Timer_A and the CPU core of the MCU are clocked by this 8-MHz clock source and generate the critical NTSC video sync and data signals. The pull-up resistor R4 and capacitor C3 provide an RC delay on the RESET pin of the MCU for proper start up when powered up.

This video game system also incorporates audio generation during a play session. This is realized by tone generation using the PWM output signal from P1.2/TA1. The PWM output is filtered by R8, R9 and C4 to generate a smooth audio signal.

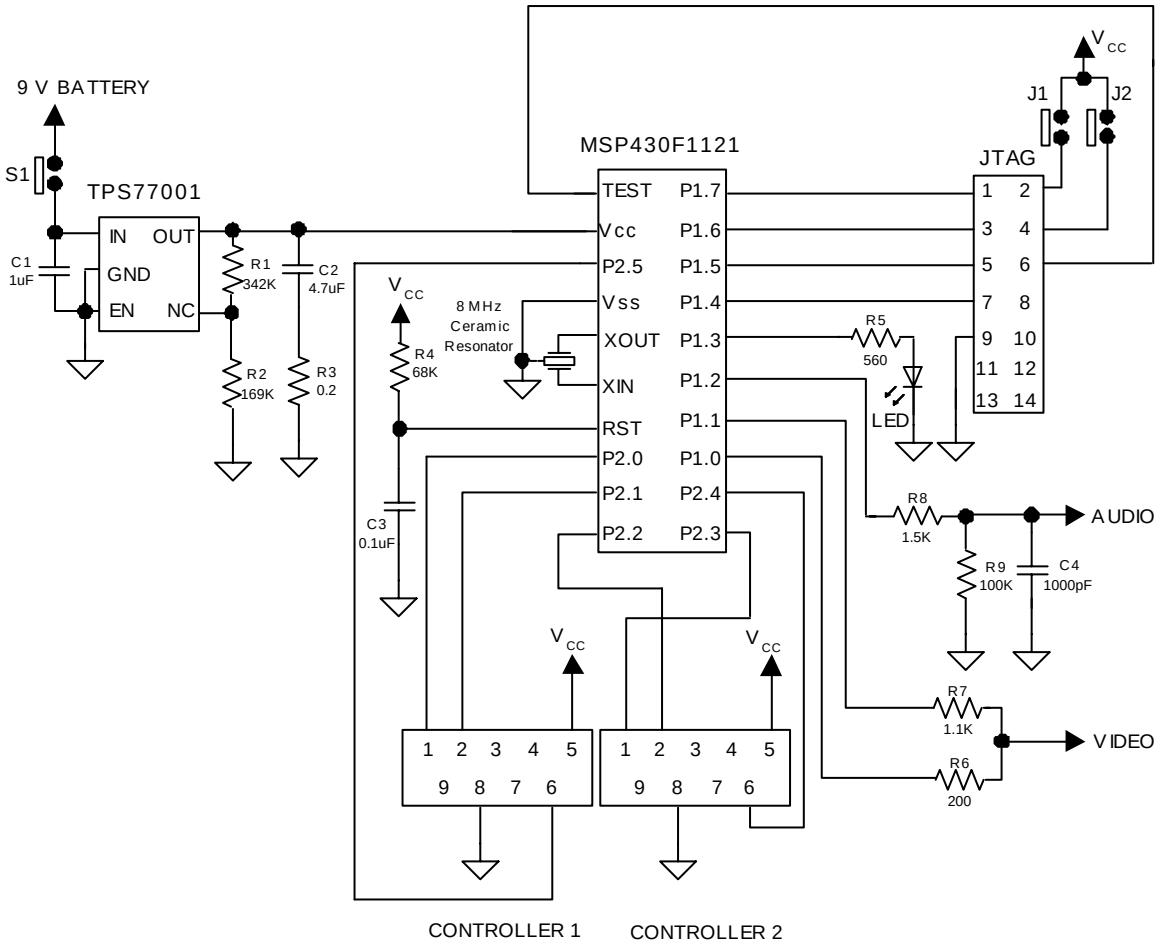

Figure 2. Video Game System Schematic

Software

The software implemented has the following functions:

- Allocate RAM for graphical data
- Program the Pong game to update graphical data in RAM
- Generate the audio and NTSC video signals in the Timer_A0 interrupt handler routine

The image data for the video output is mapped in the graphics memory allocated in RAM by the software. The screen is set up to have a resolution of 32x24 pixels. There are 32 bits of image data for the 32 horizontal pixels that are organized as 4 bytes in the graphics memory. Each horizontal pixel line is displayed on eight TV lines vertically on the screen.

All the processing for the Pong game such as checking the controller, moving the paddles and the ball are done during the upper blank scan-lines when no graphics are drawn on the screen.

The image data is output to the 2-bit DAC when a Timer_A0 interrupt is executed, which occurs every 64-µs as configured in the main.c. The Timer_A0 interrupt handler NTSC.s43 is written in assembler because of the critical timing requirements and is implemented with a jump table to generate the required signals. Video data are sent out by loading registers R11 and R12 with 4-bytes of the current horizontal line as two 16-bit words and shifting the registers bit by bit in to the carry bit and then shifting it out to the P1OUT register. In addition to this, CCTL1 is set to generate a PWM signal, which causes an audio output. The audio is created during the free cycles close to the horizontal sync pulses.

The following functions are implemented:

- Generation of the vertical and equalization pulses
- Loading the image data to the graphics memory
- Generation of the audio signal on P1.2
- Output the video data to ports P1.0 and P1.1.

Example Code

The example code for this application is available from the MSP430 website at http://www.ti.com/sc/msp430, as a zip file under this application note. The code is written both in C and assembly language using the IAR KickStart Integrated Development Environment.

The source code files are listed below:

- main.c – Initialization for the MSP430F1121A operation and the video memory set up to display the first screen on power on
  - MSP430 is set up to operate at 8 MHz
  - General I/O pins are configured for Timer_A module and for controllers operation
  - Loop that loads the image data for the first screen and calls the pong_game function

- pong_game.c – Function for the actual game where the movement of the ball and paddles are generated and loaded as image data to graphics memory
- score_board.c – Function for tracking the score of each player and loading the image data in video memory
- NTSC.s43 - Assembly language interrupt handler function for Timer_A0. This function generates the horizontal, vertical and equalization pulses, loads the image data from video memory to the output registers R11 and R12, shifts out the values to the screen and generates the audio
- msp430pong_screen.ci - Character map for the opening screen on power on
- graphics.ci – Character map for the score board and winner screen

Playing the Game

- This game system is set up for two players. The first image on the screen will show, “Pong with MSP430”. To pass this screen and to start the game press button C on either controller.
- The first player always has control of the serve. To release the ball and start the game press button C.
- The game is set up to track the scores such that the player that first reaches 3 points wins.

References

1. MSP430 x11x1 Data Sheet (SLAS241)
2. MSP430x1xx Users Guide (SLAU049)
3. TPS77xxx Data Sheet (SLVS210)
4. MSP430 Pong by Chris Liechti

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265