Spartan™-3AN Starter Kit Paint Demo

Eric Crabill

Xilinx, Incorporated

04/01/2007

Agenda

- How to set up the demonstration
- How to operate the demonstration
- Evaluating Spartan-3AN Flash
- Demo technical details

Demonstration Setup

- Make sure power switch is turned off; connect power supply to board
- Connect suitable display device
  - CRT (preferred)
  - Flat panel / projector
- Connect PS/2 mouse
- Set SW0 through SW3 to off
- Set J26 mode jumper for ISF configuration (as indicated on PCB silkscreen)
- Turn on power!

Power Jack and Switch
VGA Port
PS/2 Mouse
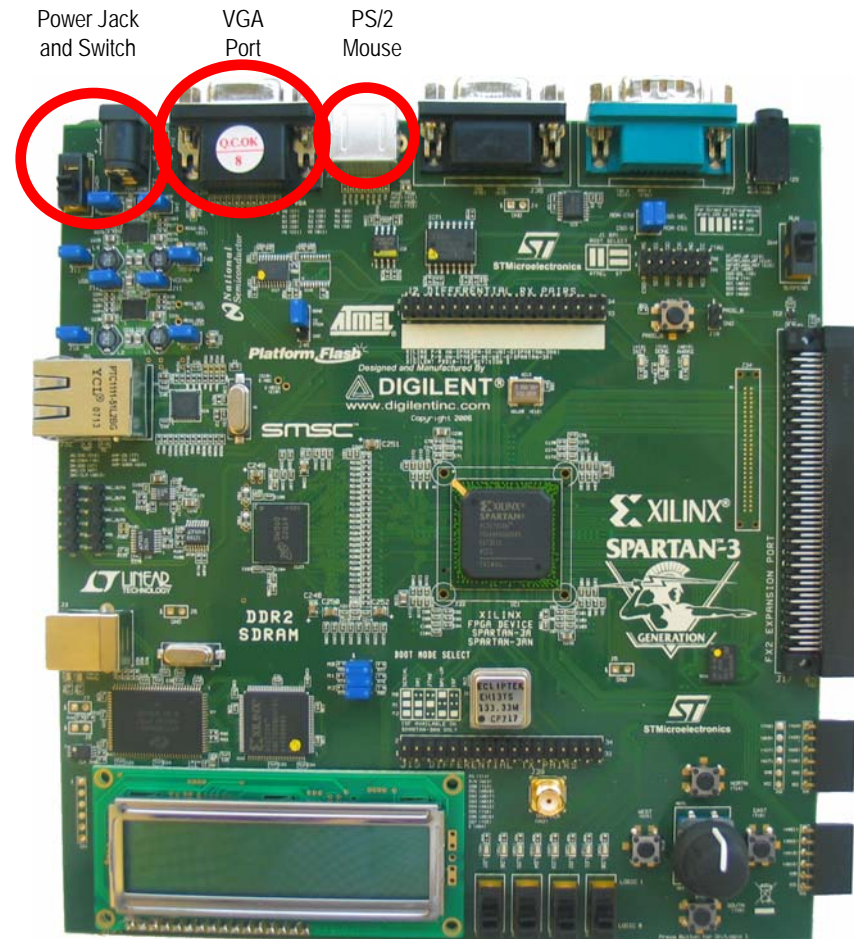
How to Operate (1)

- The display is divided
  into two regions
  - Left side is a 480 x 480 canvas
    for monochrome pictures
  - Right side displays messages,
    instructions, and simple GUI

- Use the mouse to move the
  pointer (cursor)

- Use mouse buttons to draw
  and use the GUI

How to Operate (2)

- Use the left mouse button (LMB) to draw

- Use the right mouse button (RMB) to erase

- If you have a three-button mouse, press and hold the center mouse button (CMB) while also using LMB or RMB, to use a larger "brush"

How to Operate (3)

- Activate buttons by pointing and clicking LMB
  - "Clear" will clear the entire canvas (to white)
  - "Fill" will fill the entire canvas (with color)
  - "Color" will cycle through 7 possible colors
  - "Load" buttons load a picture from internal flash
  - "Store" buttons store current picture to internal flash

- Seven picture storage locations are implemented

XILINX®
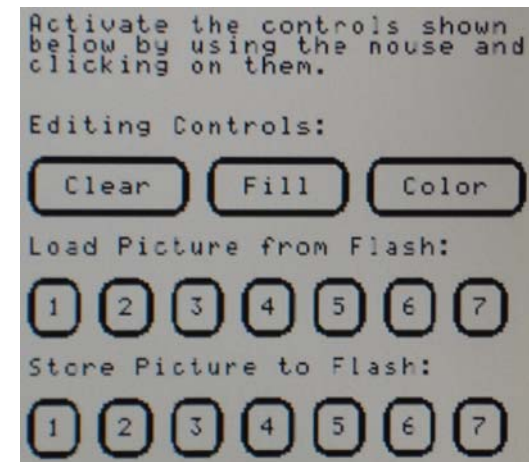
Evaluating Flash

- The initial internal flash programming includes seven pre-stored pictures
- This shows how user data may be stored after configuration data
  - Draw an original picture or edit existing
  - Store your picture to flash
  - Cycle the board power
  - Reload your picture

Evaluating Flash

- After using the demo, it may be desirable to recover the pre-stored pictures
  - Set SW0 through SW3 to on
  - Simultaneously click LMB and RMB (purposely difficult, you may need to try a few times...)
  - The cursor becomes an hourglass indicating the demo is busy (takes several seconds)
  - Set SW0 through SW3 to off
  - Original picture set is recovered!

Demo Technical Details

- PicoBlaze^{TM} processor evaluates user inputs and controls peripherals and interface to internal flash
- Frame buffer, hardware cursor, and character controller are software programmed
- Other notable features include
  - PS/2 mouse interface
  - SPI_ACCESS use

Tech: PicoBlaze

- This module was obtained from the Xilinx website, http://www.xilinx.com/picoblaze and is not included in the source download
- The primary function of PicoBlaze processor is to drive the hardware based on user inputs
  - Demo program is stored in a single Block RAM
  - Excellent "programmed" alternative to an FSM

Tech: Graphics

- Three pixel sources are implemented and multiplexed with priority
  - 16x8 hardware cursor for a mouse pointer using Distributed RAM
  - 32x64 character display for on-screen text using two Block RAM
  - 480x480 monochrome frame buffer using 14 Block RAM and some Distributed RAM

Tech: PS/2 Mouse Interface

- Simple data path with FSM controller places a PS/2 compatible mouse into "streaming mode"

- Mouse transmissions are received and buffered in a Distributed RAM FIFO specially designed to overflow without losing packet synchronization

- Software is responsible for parsing data packets and tracking pointer position and button events

Tech: SPI Interface

- SPI interface is implemented with general purpose I/O under software control
- SPI interface is identical to what might run off-chip, except that the SPI signals are interfaced to the SPI_ACCESS primitive

Common Questions

Q: "Why is the CRT preferred?" or "Why is there an interference pattern on an LCD?"

A: Most digital display devices must interpolate to display this VGA mode, the resulting interpolation errors display as an interference pattern.

Q: "Why is the picture quality / color depth low?"

A: This demo is implemented entirely in the FPGA without use of external memory.

Common Questions

Q: "Why does the flash access seem slow?"

A: The SPI interface is done with general purpose I/O, under software control. This is flexible and easy, but slow. A hardware implementation could be an order of magnitude faster.

Q: "How do I program this design into the FPGA?"

A: Use iMPACT in JTAG programming mode. Assign the xc3s700an.mcs file to the FPGA and select "program". Make sure J26 is set properly.