AppleFPGA

Users Guide for
Version 1.5

Legal Stuff

Please report bugs to the author, but before you do so, please make sure that this is not a derivative work and that you have the latest version of this file.

Introduction

AppleFPGA is a Verilog/VHDL implementation in a programmable FPGA of an unenhanced Apple][e (Apple). It gives much the functionality of the original Apple and a few upgrades. AppleFPGA version 1.5 is implemented into a Digilent Inc. Spartan 3 Starter Board. The AppleFPGA features include keyboard, video, Super Serial Card, hard disk drive, floppy disk drive, joystick, and speaker. Each feature is explained in detail in this document.

The Digilent board has almost everything needed to implement the Apple in an FPGA. An additional analog add on board is required for a few of the features including Joystick, Super Serial Card, and speaker. The Analog board is not needed for basic Apple functionality. The design files are included for this add-on board.

6502 Processor

The Apple came with a 6502 processor. AppleFPGA has a 6502 compatible CPU core. The default speed for the Apple was 1.023 MHz. The AppleFPGA runs a slightly faster speed of 1.042 MHz which is 50 MHz / 48. The Digilent Starter board includes 8 slide switches; switches 0 and 1 are used to set the speed of the CPU.

Switches		Speed	
1	0		
OFF	OFF	1.042 MHz	(Normal Apple][e speed)
ON	OFF	2.083 MHz	
OFF	ON	12.5 MHz	
ON	ON	25.0 MHz*	

*Because of the way video memory reads are interlaced with CPU accesses, a 25 MHz CPU clock cycle is skipped every 14 cycles. The effective CPU speed is 25 * 13/14 = 23.214 MHz.

The code includes a 6502 “core” written in the HDL VHDL. This code is copyrighted by Daniel Wallner. It is cycle compatible with the standard NMOS 6502. Please inspect the VHDL source code for additional details and restrictions.

ROM and RAM

The original Apple came with 16K of built in ROM, seven slots to insert add on cards, and a single AUX slot for an 80 column video / memory card. AppleFPGA also has the 16K of built in ROM. This ROM is implemented using initialized block RAM inside the Xilinx Spartan 3 FPGA. Nothing has to be loaded to boot directly into BASIC.

AppleFPGA does not support the Apple slots for adding I/O cards. Like the Apple //c, AppleFPAG has built in I/O that appears to reside in the slots. See the I/O section of this document for a list of emulated I/O cards and their slot numbers.

In addition to the built in ROM, AppleFPGA includes 2K of ROM space, $C100 - $C7FF, used for storing the ROM on the supported I/O cards and an additional 2K for the $C800-CFFF ROM for the Super Serial Card. All these ROMs are also implemented using initialized block RAMs inside the Spartan 3 FPGA.

The Digilent board includes 1Meg of high speed SRAM implemented as two 256k x 16 bit chips. One of the 256K x 16 bit chips is split into two sections. 64k x 8 bits is used as Apples main memory including the additional 16K of Language Card memory. The unused SRAM of this chip emulates an Applied Engineering RAMWORKS III card plugged into the AUX slot. The RAMWORKS III supports 4 groups of 8 memory chips. Each group can be either 256K x 1 or 64K x 1. This implementation appears to be one group of 256K x 1 and 3 groups of 64K x 1 giving the total of 448K of additional RAM space.

I/O Cards

The Apple][e came with seven slots. AppleFPGA has a function for almost every slot. Here is a complete list of the slots and their function.

Slot	Function
1	Reserved (See the clock section)
2	Super Serial Card
3	80 column text firmware
4	Clock Card
5	Empty
6	Floppy Disk Controller
7	Serial Virtual Disk Controller

The slot functions are documented in their own sections.

Keyboard

AppleFPGA uses a PC compatible PS/2 interface Keyboard to emulate the Apple ASCII keyboard. All the Apple keys are emulated including the Open-Apple and Closed-Apple keys. The keyboard does not emulate the layout of the Apple keyboard. The keys are emulated as they are displayed on the key caps. There are three special Apple keys not included in the standard PS/2 keyboard. The Open-Apple key is emulated by the Left-Alt key and the Closed-Apple key is emulated with the Right-Alt key. The RESET key is emulated with a three key combination, Control-Alt-Delete. The RESET key is also emulated with Button 4 on the Digilent board.

Because of the way the PS/2 keyboard works, the Open-Apple and Closed-Apple are not valid at power up time. This prevents the use of these keys for two Apple][e boot options. The Open-Apple and Reset combination forces the Apple to do a cold Reset and boot from the default disk drive. On AppleFPGA, use Button 0 for the Open-Apple key with Reset to force the Cold Reset. Closed-Apple and Reset are used to enter the built in system diagnostics. On AppleFPGA, use Button 1 for the Closed-Apple key with Reset to run the built in diagnostics.

Video Display

The AppleFPGA display uses a standard VGA type monitor. All the video modes of the original Apple][e are available, including the DHRES mode. The video output is set at the standard VGA resolution of 640 x 480 pixels at 60 Hz frame rate. The Apple DHRES video uses a 560 x 192. The Apple horizontal resolution has 80 less pixels than the VGA screen. Half of this pixel space is divided on each side as additional border area. Since the number of lines in the Apple Video is less than half the number, each line is scanned two times on the VGA screen. The remaining lines are also used ad additional border, but the space is not split in half. Most of the space is on the bottom and some of this space is used to display a real time clock. See the clock section for additional details.

The Digilent Spartan 3 Starter board only has 8 color capabilities. These 8 colors are the different combinations of the three primary colors, Red, Green, and Blue. A modification to the board to add an additional bit to each primary color is supported by AppleFPGA. The modification details are included in the AppleFPGA distribution.

AppleFPGA has a synthesis option to turn on the supported modifications. If this modification is not wanted, AppleFPGA will try to emulate the additional colors by modifying the on time of each color. The synthesis option to turn on the modifications is located under the banner.

```
`define SIX_BIT_COLOR
```

Place double forward slash comment character, //, in front of this line to turn the option off and automatically turn on the multi-color emulation.

Floppy Disk Drives

As previously mentioned, the original 64K memory and the additional 384K RAMWORKS III memory are held in one chip on the Digilent board. The other 256K x 16 bit chip of SRAM is used to emulate two floppy disk drives attached to the standard Apple floppy disk controller plugged into slot 6. These solid state floppies are compatible with all the Apple software that I have tried. They can be formatted with Filer or Copy II+ Version 7.4. You can use the Disk copy feature in Copy II+ to write images into the floppies. I have successfully booted from the floppy after writing DOS 3.3, Castel Wolfenstein, Zaxxon, and ProDOS images. The timing of the floppies is based on the CPU clock. This means that changing the CPU clock to a different speed does not affect

the ability to access the floppies. Also, because of this feature, the floppies run much faster when the CPU speed is set to the higher frequencies. At the highest CPU frequency, the floppy is emulating a disk where the rotational speed is (23.214MHz/1.023MHz) x300RPM = 6807.6RPM.

AppleFPGA implements a full 40 tracks on each disk. The ability to use half tracks is also supported. But just like the standard Apple Floppy, when you use half tracks, the next track needs to be at least one full track away or data will be over written. The reasoning is different, but the results are the same. Each track of the floppy uses 6312 bytes of the RAM. This gives each floppy the capabilities of 41.5 tracks.

One of the slide switches on the Digilent board, SW4, is used to swap the floppies. With the switch in the OFF position, floppy 1 is Drive 1 and floppy 2 is Drive 2. When the switch is set to the ON position, the floppies are reversed, floppy 2 becomes Drive 1 and floppy 1 becomes Drive 2. Because several programs came as two disk sets or front and back of a single floppy, this switch makes it possible to run these programs. Before trying to run this program, load floppy 1 with the first disk image and floppy 2 with the other image. When the program asks to insert the next floppy, flip the switch. If a program uses more than two floppies, then it probably cannot be run on AppleFPGA. There are two switches that are used to write protect the floppies, SW2 and SW3. The switches stay with the floppy and not the drive. There are two LEDs that show floppy activity, LED 6 and LED 7. Like the write protect switches, the LEDs are locked to the floppy and not the drive. See the section on Switches, Buttons, and LEDs for a complete details.

Serial Virtual Drive (SVD)

The main ProDOS compatible storage mechanism for AppleFPGA is a virtual hard disk drive file located on a PC server. More information on this can be obtained from Terence J. Boldt's excellent Apple II project web site, apple2.boldt.ca. The PC software, server.exe, is included in this release along with a sample hard disk drive image. The hardware and ROM for the SVD looks like a card plugged into slot 7 and uses some of the slot 4 ROM space that is not being used by the clock ROM. AppleFPGA runs this serial link at 115200 Baud. The drive appears as two 16 Meg hard disk drives in one 32 Meg file. The Apple is capable of interfacing drives up to 32 Meg. But the newest version of Copy II+ that is compatible with the NMOS 6502, has an issue with drives larger then 16 Meg. Newer versions require the 65C02.

Files can be added to the hard disk images by one of several methods. One method is to use one of the available Apple II emulators that support HDV files. Another method is to use A2 Oasis Disk Image manager. Several disk images can be created. Just like a floppy, a disk can be “changed” after the system is booted. To change a disk, the PC program should be terminated, and then restarted with the desired image. ProDOS will pick up the new image. Also, just like the floppy, be sure that the ProDOS does not have files open when the image is changed.

Super Serial Card

A Super Serial Card is also implemented in the AppleFPGA. The card appears to be plugged into slot 2. The RS232 level shifters and connector are on the analog board. The connector is a DB-9 compatible with the male DB-9 RS232 connector used on a PC. Only a few of the handshaking lines are implemented. These include CTS and RTS. The analog board is required for the Super Serial Card to work correctly. But if the Super Serial Card is not needed, then this circuitry can be omitted.

The interrupt signal, IRQ, from the Super Serial card is also implemented. The original Super Serial Card had a switch to enable / disable this IRQ. AppleFPGA uses another one of the slide switches, switch 5, for this functionality. When switch 5 is in the OFF position, the IRQ is enabled. And when it is in the ON position, the IRQ is disabled. Read the Super Serial Card documentation or the documentation for the program that you will be using with the Super Serial Card for information about the correct setting.

Clock

AppleFPGA supports a ProDOS compatible clock in slot 4. The clock keeps track of the seconds, minutes, hours, days, and day of the week. The clock is not too sophisticated. It assumes all months have 31 days. It will increment the day at midnight, but the month and year are never incremented. This means that on October 31 at midnight, the clock will become October 1. The binary for the Apple ROM is included in the distribution file. This clock only supports ProDOS mode. It either redirection command is given, IN#4 or PR#4, the clock ROM will not respond. At each entry point is a return from subroutine. The clock is set by reading the time from the PC over the same link as the SVD. The code for setting this clock is placed in the ROM for slot 1. Nothing else can be used for slot 1. Set the clock with this BASIC command:

CALL -16128

The clock uses all 16 IO bytes that are allocated to slot 4. Each byte, except for one, holds a clock data byte that is ASCII encoded.

C0C0	Year Thousands (hard coded to 2)
C0C1	Year Hundreds (hard coded to 0)
C0C2	Year Tens
C0C3	Year Ones
C0C4	Months Tens
C0C5	Months Ones
C0C6	Day of Week (0-6)
C0C7	Days Tens
C0C8	Days Ones
C0C9	Hours Tens (0-2)
C0CA	Hours Ones

C0CB	Minutes Tens
C0CC	Minutes Ones
C0CD	Seconds Tens
C0CE	Seconds Ones
C0CF	See text

The clock can also be set manually by writing the time into these values, with the exception noted in the Year Thousands and Year Hundreds. The clock will retain the time through a Reset, but it will not retain the time is the FPGA is reloaded. Also, the clock is not battery backed up, it will not retain the time when the power is removed. The byte at C0CF holds an eight bit counter, 0 to 255, which increment every 63.84 uS. This could be used as a seed for a random number generator.

The clock runs a little fast as it is based on the Vertical Sync Rate. The actual rate is 60.13 Hz. The seconds are derived from this rate divided by 60. Regularly updating the clock from the PC is the easiest way to have accurate time.

The clock is also displayed in the unused space below the bottom line of the video display. This clock display is hardware driven and shows the same time as the ProDOS clock. If the ProDOS clock is set, the clock display will show the correct time.

Joystick / Paddles

The Joystick interface is an analog circuit built onto the analog board. This board has a DB15 connector that is compatible with an analog joystick for a PC. The schematic for the Joystick interface is included in the distribution. If the Joystick interface is not needed, the additional circuit does not need to be built. AppleFPGA will run correctly without this interface connected, but no joystick will be available.

The original Apple joystick used a software timing loop to calculate the position of the joystick. Because AppleFPGA runs at different CPU speeds, using the original Apple circuitry would make the joystick unusable when the CPU is run at higher than the default speed. AppleFPGA uses a timer to calculate the positions independently. These readings are done four times a second in hardware. When the software program does the timing loop, the joystick signals the CPU sees are emulated using the last joystick hardware reading. This emulation is timed using the CPU clock to compensate for any increase in CPU speed. This allows the joysticks to be used regardless of the CPU speed.

Speaker

An additional schematic for the speaker circuit is also included in the distribution. As with the Joystick, this circuit is not needed unless speaker support is desired. This circuit was designed to power a set of stereo headphones or powered speakers.

Toggling of the speaker is done by reading from IO location C060. With an Apple][e writing to this location is not defined. But the AppleFPGA uses writes to this location to

accomplish two different upgrade features. The first feature is to write enable the ROMs. Write enabling the ROM will allow them to be modified. A read from this location will write protect the ROMs again. The second feature that is enabled with a write to this location uses the value that is written. This value sets the color of the text display. The text screen color can be changed to one of eight different colors.

Value	Color
0	Black
1	Red
2	Green (Default)
3	Yellow
4	Blue
5	Purple
6	Aqua
7	White

Make sure you read from this location after setting the text color to make sure the ROMs are write protected. By hitting the Reset button, the color will revert back to the default color. If you want a different default color, edit the applefpga.v file to change the reset value of the user_c register.

Switches, Buttons, and LEDs

There are eight slide switches on the Spartan 3 starter board. Six of these switches are used and two are not used. Here is a list of the function of each switch.

Switch	Switch OFF	Switch ON
SW0	CPU Speed*	CPU Speed*
SW1	CPU Speed*	CPU Speed *
SW2	Floppy 1 write enabled	Floppy 1 write protected
SW3	Floppy 2 write enabled	Floppy 2 write protected
SW4	Floppy normal	Floppy swapped
SW5	IRQ Enabled	IRQ Disabled
SW6	Not Used	
SW7	Not Used	

*

SW0	SW1	CPU Speed	
OFF	OFF	1.042 MHz	(Normal Apple][e speed)
ON	OFF	2.083 MHz	

OFF	ON	12.5 MHz
ON	ON	23.214 MHz

There are 4 push buttons on the board. Three of the buttons are used. Here is a list of the buttons and their function.

Button	Function
0	Open Apple
1	Closed Apple
2	Reserved
3	System Reset

There are 8 LEDs on the board. All of these LEDs are used. Here is a list of their functions.

LED	Function
7	Slot 7 active
6	Floppy 2 being accessed
5	Floppy 1 being accessed
4	Apple ROMs write enabled
3	Floppy controller PH_3
2	Floppy controller PH_2
1	Floppy controller PH_1
0	Floppy controller PH_0

The 4 seven segment displays will permanently display A][E.

Known Bugs and Limitations

Since the CPU core is not compatible with the 65C02, only software that is compatible with the unenhanced Apple][e or earlier can be used. A lot of the available software has newer versions for later machines. This software cannot be run.

The floppy emulation is not compatibility with the bit image files available on the Internet. These images incorrectly use 6656 bytes for each track. With a disk rotating at 300 RPM and the Apple CPU writing a byte every 32 CPU cycles, the maximum number of bytes would be approximately 6400 bytes. Self sync bytes would lower this value even more. The AppleFPGA floppy is a byte oriented interface and compensates for but does not emulate self sync bytes. Because of this, the total number of bytes for each track is set to 6312. Disk drive alignment programs will report that the disk is running a little fast, this is to be expected, since the self sync bytes do not take the additional bit times.

Any new bugs should be reported to the AppleFPGA groups at Yahoo Groups. In this group, you can find the latest information and newest version of AppleFPGA.