FPGA NES

Submitted by:
Art King
art.king@ufl.edu
352-263-1403
# Table of Contents

Table of Figures ........................................................................................................... 2  
Project Summary ........................................................................................................... 3  
Background .................................................................................................................. 3  
Technical Objectives .................................................................................................... 4  
Concept / Technology Selection .................................................................................... 4  
Gantt Chart ................................................................................................................... 8  
References .................................................................................................................... 9

## Table of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>Abbreviated FPGA NES system-level block diagram.</td>
<td>4</td>
</tr>
<tr>
<td>Figure 2</td>
<td>Nintendo Entertainment System block architecture.</td>
<td>5</td>
</tr>
<tr>
<td>Figure 3</td>
<td>Proposed Core Module and partial Audio/Video Module architecture.</td>
<td>6</td>
</tr>
<tr>
<td>Figure 4</td>
<td>Proposed Power Module architecture.</td>
<td>7</td>
</tr>
<tr>
<td>Figure 5</td>
<td>Gantt chart breakdown of key milestones.</td>
<td>8</td>
</tr>
</tbody>
</table>
PROJECT SUMMARY

The goal of this project is to design a stand-alone emulator of the Nintendo Entertainment System video gaming platform whose major architecture is based primarily around a modern programmable logic device in conjunction with additional external peripheral components. The design will include a FPGA core developed in VHDL, video and audio output, and human interface controller.

BACKGROUND

Emulation of the Nintendo Entertainment System is a popular pastime amongst hobbyists and hackers alike. However, the bulk of community effort has been placed on software emulation—without a doubt driven by the relative accessibility of software development tools vs. a hardware approach—relying on observations made during community reverse engineering efforts to incorporate “soft hacks” that replicate the many nuances and quirks of the original system, let alone basic functionality. Furthermore, the multiple layers of abstraction that software emulators must pass through introduces real-time errors that are difficult to compensate for without direct low-level register manipulation. The approach this project will take is to use existing reverse engineered documentation available throughout the web to synthesize a low-level emulator on a FPGA and to design all the necessary peripherals needed to support such a design.

As of writing, there are four known projects that have successfully emulated the NES on a FPGA platform (in order of discovery):

- Kevin Horton – developed on a custom 2-layer Altera Cyclone I platform[1].
- Dan Leach – developed in VHDL on a Altera UP3 development board as part of a Master’s project at Bradley University[2].
- Dan Strother and Brent Allen – developed in VHDL as undergraduates at Washington State University; Dan has since expanded the project with numerous enhancements on a Digilent Nexys development board based on a Xilinx Spartan-3E and has ported a large section of his original code to Verilog[3].
- Jonathon Donaldson – an active and successful SystemVerilog emulation project-in-work with focus on HDL development rather than hardware[4].

This project will differ from the above known attempts in that 1) a modular prototype will be consolidated into a single multi-layer design whilst facilitating future expansion and development, 2) a modern 60-nm technology node Altera Cyclone IV FPGA will serve as the system’s programmable core, and 3) external memory will be used to overcome embedded block RAM limitations.
**TECHNICAL OBJECTIVES**

The intent of this project is to encapsulate multiple facets of electrical engineering in a single design, emphasizing techniques in FPGA development, PCB design for high-speed signal integrity, microcontroller integration, and analog design. Figure 1 depicts a basic system-level block diagram of interaction between major components. Since a modular design has already been prototyped and tested, the next goal will be to attend to previous unforeseen limitations and consolidating the segmented design into a single PCB unit, with the only external devices needed being game controllers, AC-to-DC power supply, display and speakers.

![Abbreviated FPGA NES system-level block diagram.](Image)

**CONCEPT / TECHNOLOGY SELECTION**

The Nintendo Entertainment System functions around two main chips: a Ricoh 2A03 and Ricoh 2C02[5], as shown in Figure 2. The 2A03 can be thought of as the “brains” of the system, an ASIC containing a MOS Technology 6502 processor with its binary coded decimal arithmetic functions disabled and a custom pseudo-Audio Processing Unit. The 6502 is responsible for processing data stored in an external cartridge, interpreting controller input, and sending image data to the 2C02; the pAPU is responsible for producing five analog audio channels—two pulse waves, one triangle wave, one white noise, and one differential PCM) which are mixed to produce all game sound. The 2C02 is a custom chip developed by Ricoh dedicated to processing image data received from the 6502 and produces 256×240 pixel resolution video in NTSC format; the actual displayed image is 256×224 as a byproduct of NTSC formatting removing the upper and lower 8 scanlines. Due to the limited address map of the 6502, cartridges were
equipped with a memory map controller for bank switching in the case where the game being played required more than the 32KB of program ROM that the 6502 could map to at any given time.

The highlighted blocks shown in Figure 3 will compose the core of the system. An Altera Cyclone IV FPGA was chosen to implement the bulk of the architecture. Despite being comparatively more expensive than an equivalent or better offering from Xilinx (in particular, the Spartan-6 class), the Cyclone IV was chosen due to availability of development tools. Since the FPGA is SRAM-based, and due to the required internal block RAM needed to synthesize the architecture, a 4Mbit configuration device will be integrated into the design, enabling both in-circuit serial programming of the configuration device for automatic power-on configuration and direct JTAG programming of the FPGA to speed up development. It is the author’s intent to eventually migrate the completed architecture to reap the benefits of a Xilinx Spartan-6, however, that endeavor is beyond the scope of this project’s objective.

Figure 2 Nintendo Entertainment System block architecture.
The T65 open source project available on OpenCores.org[6] will be used as a synthesizable basis for the required 6502 processor. Of the numerous synthesizable 6502 cores available across the internet, the T65 was chosen based on forum feedback as being the most reliable open source VHDL implementation. Despite praise from the online community, the implementation is in fact ridden with a handful of relatively minor bugs and features that are unnecessary; in the case of binary coded arithmetic functions, undesirable altogether. It will therefore be necessary to isolate and correct these bugs, and remove unwanted functions prior to system integration in order to achieve the closest emulation possible. In addition to the T65 core, a custom PPU and pAPU will need to be developed, along with at least one of the more widely used Memory Map Controllers. A script must be written to convert a *.nes ROM file to *.mif file which can be loaded into internal block RAM instantiations prior to synthesis. This script has already been implemented in MATLAB and is noted for the sake of documentation completeness.
The power section shown in Figure 4 will consist of the appropriate step-down and regulation circuitry needed to power all devices within the system. This segment of design is critical in that it must be capable of sourcing all voltage levels necessary for the main FPGA and all peripheral components. A 5V, 2.5A wall switching supply will provide main power to be stepped down and regulated. Although it not expected that the system will require more than 1.5A at any given time for this design, a 2.5A main source was chosen to allow for future expansion. The first stage regulator is a National Semiconductor LM2853 Buck regulator which will step down 5V to 3.3V for use by the FPGA I/O banks, microcontroller, and Bluetooth module. This Buck regulator is capable of providing as much as 3A and will be sufficient to handle future peripheral expansion. The 3.3V output from the LM2853 will then be sent to a National Semiconductor LM26420 dual Buck regulator. The first channel will step down 3.3V to 1.2V for FPGA core switching. The second channel will step down 3.3V to 2.5V explicitly for the FPGA’s analog PLL rail, which must be powered regardless of its use. Each channel is capable of providing 2A of current, with the higher load expected to come from the 1.2V channel. Both regulators switch at a nominal frequency of 550kHz and incorporate synchronous rectification, providing a reasonable compromise between reducing board real estate and maximizing efficiency. Due to the inherent sensitivity of switching regulators, a high level of attention to detail must be placed on PCB to ensure clean power and to minimize the effects of electromagnetic interference, in particular, from the digital devices on the core section of the system. It will also be necessary to filter rails which will provide power for sensitive segments, in particular, the 2.5V analog PLL rail used by the FPGA.

The audio/video section shown in Figure 1 will be designed to produce video output from the digital data processed by the synthesized PPU. The original modular prototype implemented VGA and a 6-bit R-2R ladder D/A converter per channel using precision resistors matched to
VGA input impedance specifications; sync lines were also level-shifted from 3.3V to 5V using MOSFETs. Since this aspect of the design proved successful without fault, the design will be migrated towards a more modern display standard; DVI is the current candidate. The specifications for audio output will be left fairly ambiguous for the time being as it requires further development of the custom pAPU architecture in order to determine exact requirements. At a minimum, digital sound in the form of PWM generated by the FPGA will be filtered, mixed and amplified prior to being delivered via 3.5mm TRS jack using single-supply op amps to enable either headphone or portable speaker output. This portion of the design will rely on the application of analog principles in order to achieve the best sound quality.

**GANTT CHART**

![Gantt Chart](image)

The Gantt chart shown in Figure 5 identifies target milestones to be achieved within the limited time frame. Since architectural research is complete and most of the design has been successfully synthesized, focus for the first month will be placed on refining the image rendering pipeline and designing the custom pAPU architecture. Hardware additions will include incorporating external RAM to compliment the FPGA’s insufficient internal BRAM; BRAM will be used primarily for frame buffering. External ROM will be incorporated on the microcontroller end, which will act as a direct replacement for a physical cartridge. An alternate gamepad will also be incorporated. Due to unforeseen limitations of existing embedded Bluetooth modules on the market, the Wii controller integration previously attempted will be replaced with an alternate device.
REFERENCES

   Internet: kevtris.org/Projects/console/sections/, [Aug. 25, 2011].

   Internet: cegt201.bradley.edu/projgrad/proj2006/fpganes/, [Aug. 25, 2011].

   Internet: danstrother.com/fpga-nes/, [Aug. 25, 2011].

