6710-6
Andrew Bradbury
Steven Brown
Tim Grant
Travis Gray

Project Proposal

Our plan is to design a microprocessor modeled after the CR16 microprocessor. The instruction set will follow those given in the ECE 3710 course, and will use the same instruction encoding. The programmer's guide that complements this ISA will be used to ensure proper execution of each instruction. This will allow us to have some general purpose use for this chip.

Our design includes a sample-based audio playback system that operates largely independently of the main processor. The audio system has access to dedicated sample memory and a DAC, both of which reside off-chip.

Because memory is the largest real estate consumer for our design, we plan to move all of the memory off chip. The challenge with this is to make sure that timing is not compromised for memory accesses. For this reason, we have designed our system around a 25MHz clock, and will use memory chips that are 2 bytes (one word) wide. The 25MHz clock will work nicely with a VGA display, and the word-addressed memory will allow us to match timing specifications.

Target applications for this chip include a drum machine and simple video games. The chip will include input that will connect to gaming controllers, and outputs for audio and graphics. This system structure will allow for reuse of the machine for games and other applications that follow the programmer's guide.

The core of the chip is a 16-bit processor. We are taking a modular approach to the design of this processor, using behavioral Verilog to synthesize some modules (e.g., the controller and the decoder), and a mix of custom cells and structural Verilog to construct the ALU and the datapath.

We are implementing a VGA interface for graphical output. The VGA interface will be implemented using behavioral Verilog for the state machines, and possibly using custom counter circuits, depending on size constraints. The VGA controller is glyph-based in order to reduce computational and memory requirements.

Our design features an audio sub-system that is largely independent of the processor. The audio system consists of:

- multiple address generation units, which determine the addresses of audio samples that need to be fetched from the flash memory;
- a unit to sum sample streams (using integer arithmetic);

- a SPI interface to stream sample data from off-chip flash memory;
- and an I2S interface to stream the summed samples to the off-chip audio DAC.

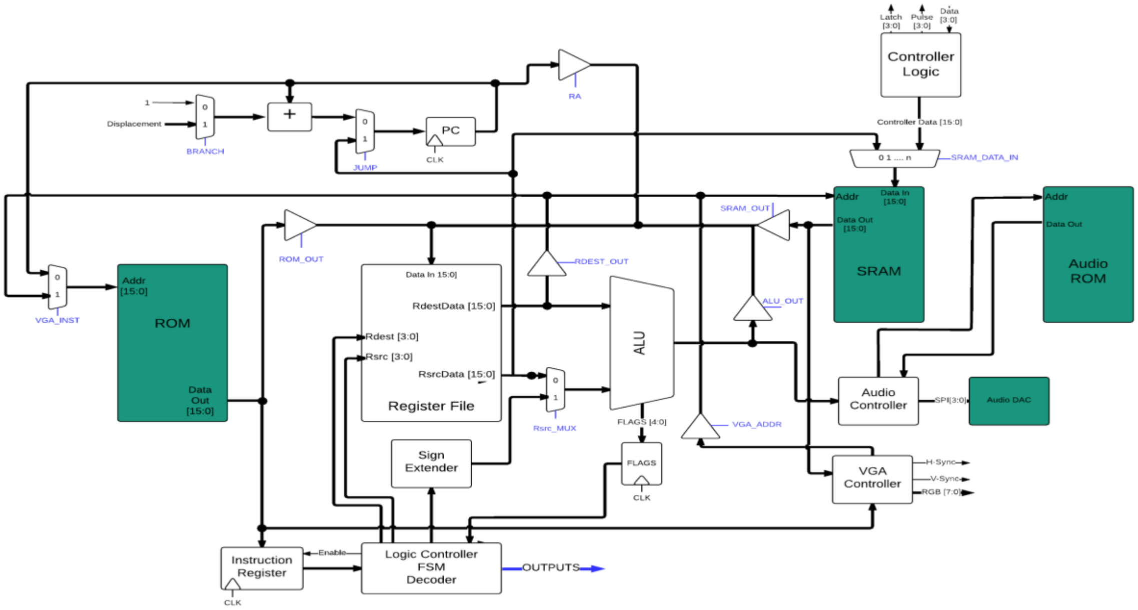
These units will be implemented as state machines using behavioral Verilog. The counters and comparators used by the address generation units may need to be custom designs, depending on how well the synthesis tool is able to translate our behavioral Verilog designs.

Our design makes extensive use of off-chip memory. Program instructions and the VGA glyph library are stored together in a 64k x 16-bit ROM, which shares data and address busses with a 64k x 16-bit SRAM used for runtime storage. The audio data is stored in a separate off-chip flash memory and is accessed by the audio system via a dedicated SPI interface.

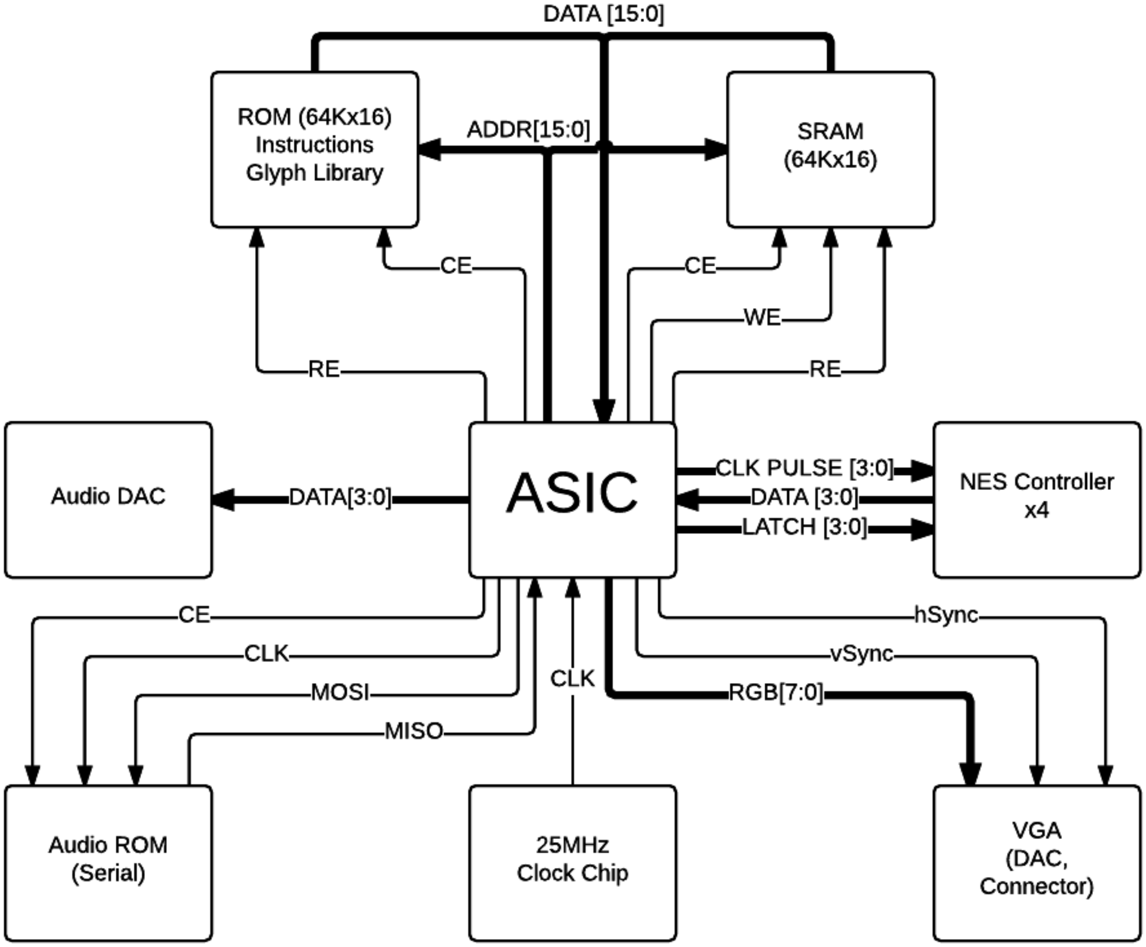
The on-chip storage requirements include a register file for the processor, various peripheral status registers, FSM states, and potentially a block of SRAM to be shared between the processor and the audio system.

We will probably need to add a few additional cells to our library to meet our design goals. We have already implemented a full adder and XOR cells to help meet design goals, but it may be necessary to modify these cells or add others depending on whether custom circuitry is needed to stay within the area constraints. We may need a comparator cell, as well, for comparisons that need to be performed outside of the ALU (e.g., in the VGA controller and in the audio system). Since our design includes serial interfaces (SPI and I2S), including an assortment of shifters might be in our best interests.

We’ve included both a system diagram and diagram of the CPU design:



The system diagram is as follows:



The chip will have 84 total pins, utilizing four TCUs. Of the 84 pins, 6 will be dedicated to power and ground. This leaves a total of 78 signal pins available for signal I/O. Of these 78 pins, 68 pins will be used for ROM, SRAM, SNES controllers, VGA, and audio. This setup gives the desired versatility for multiple applications.

We have determined that 4 TCUs are needed to ensure we have enough pins to support off-chip memory and meet the other I/O requirements.

- 12 pins will be used for SNES controllers
- 16 pins for the memory address bus, shared between the SRAM and ROM
- 16 pins for the data bus that again will be shared between the SRAM and ROM
- 5 pins for enabling read/write for the SRAM and ROM
- 4 pins for the audio ROM SPI interface
- 4 pins the DAC I2S output
- 8 pins VGA RGB
- 2 pins for VGA VSync/HSync
- 1 pin for external clock input

A more complete description of the pins and their uses can be found in the block diagram of the system. With the remaining 10 pins, the team will determine which internal signals will be most useful for testing the fabricated ASIC.

In order to test the chip we are going to take two separate approaches. We plan on fabricating a PCB test board that will enable us to test the target applications for which we are designing the chip. However, due to the cost of the board, the components required to connect to external elements (VGA/DAC/Controllers), and potential errors in the PCB, we plan on fabricating only one board, and testing the rest of the chips using the tester.

Our custom board will enable us to test the controller inputs, VGA, and the audio circuitry. Building this board will also ensure that our data path is functioning as expected because it will need to interface with our selected RAM and ROM chips. On this board, we will provide pin headers to allow us to observe the state of certain systems in our chip. The signals that will be pulled out to the extra pins are the four FSM state bits, and the five ALU flag bits. THis will allow verification that the control FSM and ALU are operating properly. The fact that the external data busses, for the SRAM, are driven by the ALU will allow us to observe its output as programs execute. This will allow us to test the functionality of portions of our chip in the event that the system as a whole does not function correctly.

We do plan on fabricating our chip. Steven Brown and Travis Gray will take the class in the spring, and possibly other members of this team as well.