David Levenson
Danny Malconian

# Digital Percussion and Entertainment System Project Proposal and Block Diagram

The digital percussion and entertainment system is an interactive drumming and video kit. The system can be used in two ways. The first way is to simply allow the user to drum on the kit and have the appropriate drumming sounds be played back from the system. There will be preprogrammed drum noises in memory corresponding to the different pads available to hit on the system. Secondly, there will be a video monitor which will display a game the user will have to drum to. The game will display a character on a map. The character will have to navigate a map in a certain amount of time. The character's speed will be determined by the speed at which the drum kit is hit and there will be a separate pad to allow the character to jump. The user will be able to select from two modes (playback or character movement game) after starting the system or resetting the system. The sound output will be through speakers and the video will use xvga.

We will implement this system in four major sections. The four sections are the drum input section, the playback section, the game controller section and the main controller section. The drum input section consists of the analog to digital converter and the hit recognition module. The playback section consists of the playback module, its block RAM, the audio module and the ac97 modules. The game controller section consists of the game controller module, the game block RAM, the divider and the timer. And finally there is a main controller section which consists of the controller module and the video modules.

The first section will be to convert the analog signal from the drumkit into a digital signal and then to convert the signal into the appropriate pulse to send to the rest of the system. This will be done with an analog-to-digital converter. After the drum is hit there will not be a sharp pulse which will easily identify the intensity of the hit. The signal will be bouncy and rounded and must be converted into a readable signal by the FPGA. The digital signal must then be analyzed to determine intensity, number of hits, and when the pad was struck. This will be done in the hit recognition module.

In playback mode, the signal from the hit recognition module will be sent to the playback section. The playback module will take the pad number input from the hit recognition module and the intensity signal to determine what sound should be outputted and retrieve the sound from a block RAM. The retrieved data will then be sent through the audio module at the correct sampling rate and the ac97 module to the speakers. In the game mode the hit recognition output signal will be sent to a game controller module.

The next section is the game controller section. If the character movement game is selected then the hit recognition signal will be sent to the gaming module again. The characters and the backgrounds and objects in the game will be stored in the block RAM connected to the gaming module. Every time a hit is received from the hit recognition module the character will move appropriately. The screen will be updated at the negative edge of the vsync signal from the xvga module. The user will have to drum fast enough to finish a level in a certain amount of time to win that level. Therefore this game will use the divider and timer modules connected to the game controller.

The divider module will alert the timer after a certain amount of time has passed. This amount of time will most likely be in the microsecond to millisecond range. There may be individual dividers for the beat response game and the character movement game or they may use the same module. The divider will connect to a timer module which will keep track of the time needed for each game.

Finally there is the main controller section. This section will allow the user to select which feature of the system they would like to use and will interface with all of the required video modules to output images to the screen. The controller will present a menu on the monitor that the user will use to select which feature to utilize. It will output which mode the system is selected to be in to the playback module, the hit recognition module and the game controller module. It will also be used to select which set of sounds to be outputted for the simple playback mode. The main controller module will handle all of the video signals as well. It will take inputs from the xvga module connected to the 65 MHz clock to output the video correctly. This module will take video inputs from the game controller module to output to the monitor and it will have its own memory for its main screen which is used to select the mode.

There will be a 65 MHz clock connected to all of the modules to time everything correctly. There will also be a global reset button to reset the system to the original menu screen. This menu screen is the screen which will show the user which functions they can select. There will be a switch input connected to the main controller module to select the function for the system as well as an enter button to enter that function. All switches will be debounced and synchronized and all button inputs will be synchronized.

If there is time we will also implement a beat response game. If the beat response game is selected then the game module will output a pad number to hit and begin a timer. When the correct signal is inputted from the hit recognition module the timer will stop and the game controller module will assign the appropriate score for that hit. The game controller module will send two signals to the playback module. The first signal will be to select the type of sound to be played and the second will be a signal containing the information encoded in the hit recognition signal. The hit recognition signal is sent to the gaming module first to keep the timing correct. The beat response game will output numbers and graphics to the screen using a block RAM attached to the module. The screen will be update at the negative edge of the vsync signal from the xvga module.

**<u>Specifications</u>**:

<u>Drum Pad ADC:</u>

*Inputs*: Analog signals from drum pads
*Outputs*: Digital signals to Hit Recognition Module
Pad number hit to Hit Recognition Module

<u>Hit Recogntion Module:</u>

*Inputs*: Drum pad intensity signal from the ADC

Pad number from the ADC
Mode Select from the main controller module

*Outputs*: Analyzed intensity signal to playback and game controller modules
Pad number to playback and game controller modules

Playback Module:

*Inputs*: Analyzed intensity and pad number from Hit Recognition Module
Ready signal from audio and ac97 module
Analyzed intensity signal and pad number from game controller module
Kit selector from main controller and game controller
Sound data from playback BRAM
Mode select from main controller module

*Outputs*: Audio sample data for output to audio module
Address and enable signals to playback BRAM

Game Controller Module:

*Inputs*: Analyzed intensity signal and pad number from Hit Recognition Module
Mode select from main controller module
Signal from main controller module indicating a new frame is ready
Video data from game controller BRAM
Expired signal from timer
Mode select from main controller module

*Outputs*: Analyzed intensity signal and pad number to playback module
Address and enable signals to game controller BRAM
Kit select signal to playback module
Video output data to main controller module
Start timer signal to timer
Time parameter to time parameter selector module

Time Parameter Module:

*Inputs*: Parameter from game controller module

*Outputs*: Value to timer

Timer Module

*Inputs*: Start timer signal from game controller module
Value from time parameter selector module
Divider clock signal

*Outputs*: Timer expired signal to game controller module

Divider

*Inputs*: Clock

*Outputs*: Specified time signal to timer module

Main Controller Module

*Inputs*: XVGA data
Video RGB data from game controller module
Video data from main controller memory

*Outputs*: Mode Select signal to Hit Recognition, Playback and Game Controller modules
New frame signal to game controller module
Kit select signal to playback module
Address and enable signals to main controller BRAM
RGB pixel data to monitor

Peripherals the User can use/see:

*Inputs*:

Drum pads: These will connect to an analog to digital converter to make the signal usable by the FPGA. These inputs will then go through the hit recognition module to assign an intensity to them. From there they will go to either the playback module or the game controller module for use in playback or in the video game if it is specified.

Mode Select Switches (2): These switches will be used to select whether the playback mode, the beat response game mode or the character navigation game mode are selected. These signals from these switches are debounced, synchronized and sent to the main controller module. Since there are only three possible modes we will only require two switches.

Function Select Enter Button (1): This button will be used to start the system in the mode specified by the mode select switches. It will be connected to the main controller module after being synchronized with the system clock. The main controller module will display a screen with all of the possible modes and will only advance into one of the modes if this enter button is pressed.

Reset Button (1): This reset button is global and will reset all modules to their initial states. This button will be synchronized to the system clock.

Kit Select Switches (3): These switches will be debounced, synchronized and then sent to the main controller module to select which set of sounds will be outputted for playback mode.

*Outputs*:

Pixel Data: This data will serve as the video output from the main controller module. It will be encoded with red/green/blue (RGB) data which will be transmitted to the monitor.

Speaker: Takes audio data from ac97 module to output

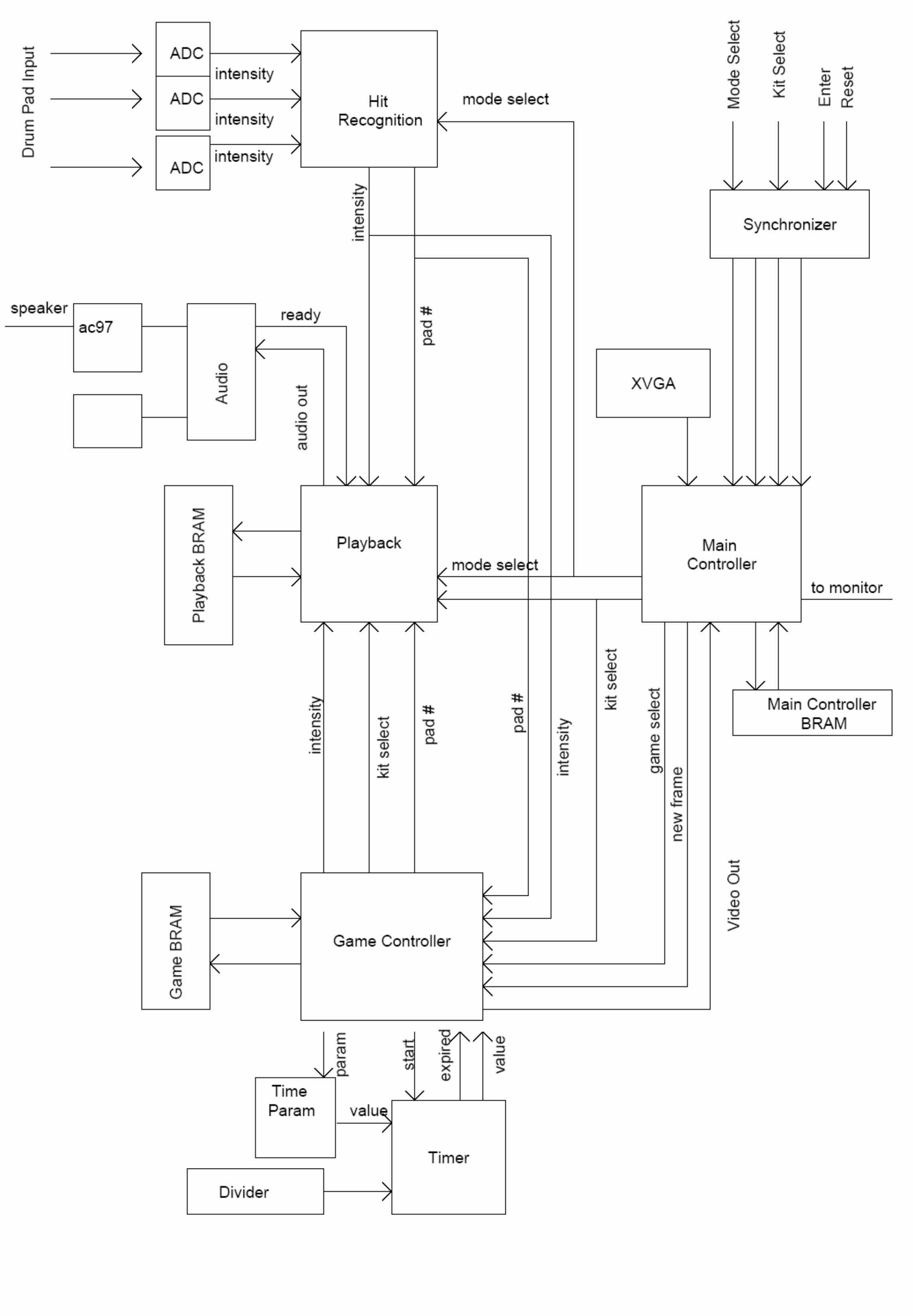
## Testing

Each module will be built and individually tested using the testbenches of the xilinx toolchain. Modules will then be connected one by one and retested with each consecutive connection. Upon downloading the program to the FPGA the logic analyzer will also be used for testing. The final test will be the demonstration of each mode of our system.

## Division of Labor

Danny will be responsible for the hit recognition module, the logic for the gaming portion, and the ADC from the drum pads.

David will be responsible for the playback module and memory, the main controller module and the video output from the gaming system.



Block Diagram