SD Card Read/Write with the Nexys4 FPGA Platform

Jono Matthews | MIT EECS | 6.111 Fall 2015
**Common Uses in Projects**

**Video**
Large color images

**Audio**
High-quality sound effects
Full-length songs

**Misc.**
Physiological recordings (ECG, EEG)
....
input  SD_CD
output SD_RESET
output SD_SCK
output SD_CMD
inout SD_DAT[3:0]

Vivado Constraints
The `sd_controller` Module: Pins and Clock

```verilog
module sd_controller(
   input reset,       // Resets controller on assertion.
   input clk,         // 25Mhz clock.
   output reg cs,     // Connect to SD_DAT[3].
   output mosi,       // Connect to SD_CMD.
   input miso,        // Connect to SD_DAT[0].
   output sclk        // Connect to SD_SCK.
);

   // For SPI mode, SD_DAT[2] and SD_DAT[1] should be held HIGH.
   // SD_RESET should be held LOW.

   ....
```
The sd_controller Module: Reading

....

output ready, // HIGH if the SD card is ready for a read or
// write operation.

input [31:0] address, // Memory address for read/write operation. This
// MUST be a multiple of 512 bytes, due to SD
// sectoring.

....

input rd, // Read-enable. When [ready] is HIGH, asserting [rd]
// will begin a 512-byte READ operation at
// [address]. [byte_available] will transition HIGH
// as a new byte has been read from the SD card.

The // byte is presented on [dout].

output reg [7:0] dout, // Data output for READ operation.
output reg byte_available, // A new byte has been presented on [dout].

....
The **sd_controller** Module: Writing

... 

**output** ready,  
// HIGH if the SD card is ready for a read or  
// write operation.

**input** [31:0] address,  
// Memory address for read/write operation. This  
// MUST be a multiple of 512 bytes, due to SD  
// sectoring.

... 

**input** wr,  
// Write-enable. When [ready] is HIGH, asserting  
// [wr] will begin a 512-byte WRITE operation at  
// [address]. [ready_for_next_byte] will transition  
// HIGH to request that the next byte to be written  
// should be presented on [din].

**input** [7:0] din,  
// Data input for WRITE operation.

**output reg** ready_for_next_byte,  
// A new byte should be presented on  
// [din].

...
A Useful Tool: HxD

Raw disk editing
No filesystem needed

Windows:
http://mh-nexus.de/en/hxd/

Mac:
http://www.macupdate.com/app/mac/17562/hexedit
Playing **Audio** with the **Nexys4 FPGA** Platform

Jono Matthews | MIT EECS | 6.111 Fall 2015
Nexys4 Pins

\texttt{ampPWM} (audio PWM signal)

\texttt{ampSD} (keep asserted whenever sound should be playing)
The audio_PWM Module

```verilog
module audio_PWM(
    input clk,               // 100MHz Clock.
    input reset,             // Reset assertion.
    input [7:0] music_data,  // 8-bit music sample.
    output reg PWM_out       // Connect to ampPWM.
);
```