NumberOneRealTimeSpectrumAnalyzerMAX

-or-

NORTSAM

Tecuan Flores, Josh Mackler, Phil Coakley

Interfacing with the AK4565 Audio Codec

Timing diagram:

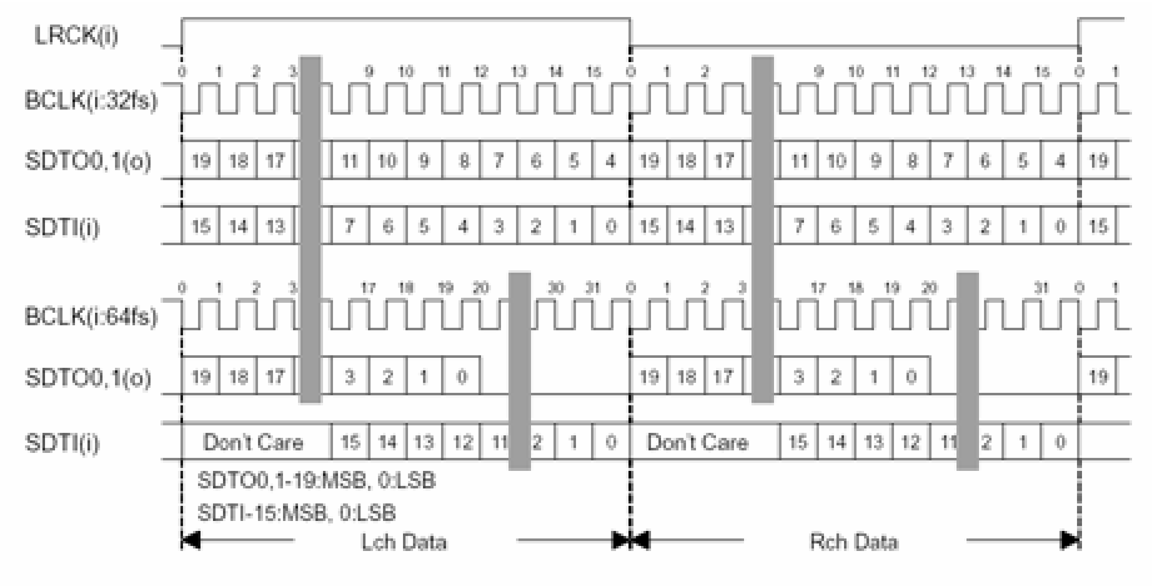


Figure 9. Audio Data Timing (No.0)

Signal definitions:
LRCK – left/right clock; when high data is for left channel, low for right channel
BCLK – bit clock; used to synchronize serial communications with chip
SDTO0 – serial output from chip to FPGA (encoding of analog in)
SDTI(i) – serial input from FPGA to chip (for analog out)

Whether we decide to use a 64fs or 32fs bit clock will depend on the details of the FFT. In any case, the circuit here is a simple shift register that puts another 20 bits into a particular memory location every time the left/right clock transitions. A few clock dividers will be needed to generate the appropriate synchronization signals.

FFT with the FPGA

Spectrum Range

The spectrum is calculated over the range of 20 Hz to 20 kHz, the accepted range of human hearing. Each band represents a third of an octave, resulting in a spectral display slightly over ten total octaves.

Calculating the Spectrum

The Fast Fourier Transform (FFT) is a simple, computationally realizable, transform that we will use to obtain the necessary frequency band. A FFT uses the structure found in Discrete Fourier Transforms to enable the use of efficient algorithms. Normal DFTs have computational complexity of $O(N^2)$, where FFTs are on the order of O(N log N), with a minimum number of multiplies and adds.

An unfortunate limit to the FFT is the linear nature of the frequency bins in the computation, instead of logarithmic, which would be more suited for audio. Consequently, a large amount of samples is needed to obtain the appropriate third octave bands. Depending on size limitations, the FFT will compute 2048 or 4096 different bins. The bin for a particular frequency is calculated by taking the size of the FFT multiplied by the frequency of the band divided by the sampling rate.

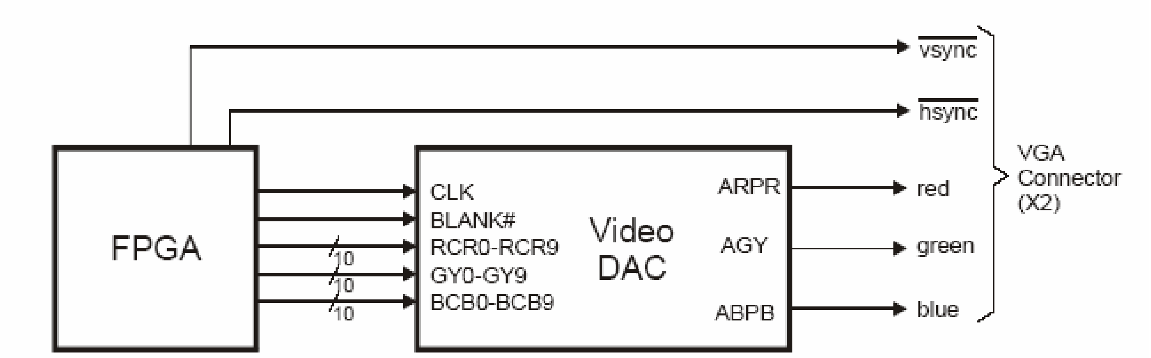
The goal is to implement the FFT completely in hardware. Adjacent bins will be added for each third octave, so that the bands can be found on the fly. Due to the linear nature of the FFT, the higher bands will take more computation, since more bins will be a part of that octave range.

Although some accuracy will be lost, fixed point multiplication will be used for the few multiplies necessary, as well as fixed point addition.

Time allowing, other, more computationally complex, methods of computing the spectrum might be pursued. In particular, there exists a constant Q transform, where the bands are based on octaves.

Interfacing with the THS8133B Video DAC

The FPGA uses a Texas Instruments video DAC (THS8133B) to generate the video signals for a VGA display. The FPGA passes 30-bit pixel data (10 bits for the red, green and blue color components) to the video DAC on each clock edge. The DAC generates the analog red, green and blue signals for the VGA display while the FPGA generates the horizontal and vertical sync pulses directly. The FPGA lowers the BLANK# signal when the pixels fall outside the desired visible area of the display.



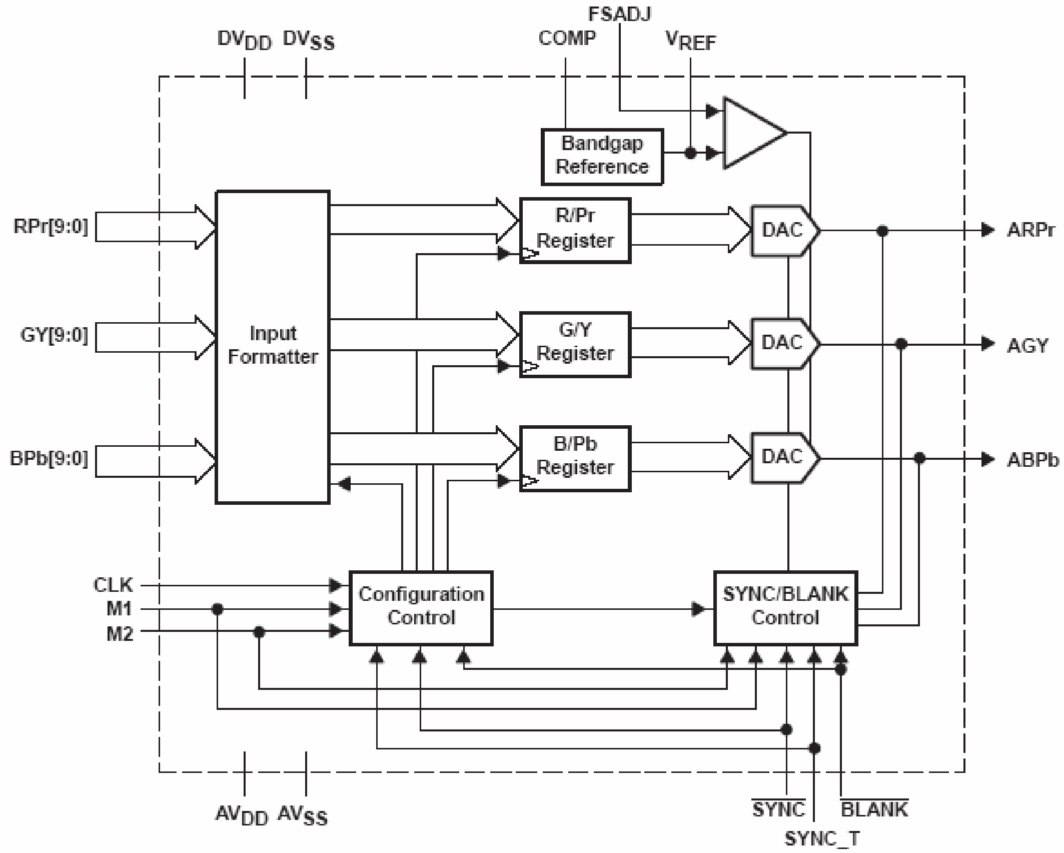


Figure 1. THS8133 Block Diagram

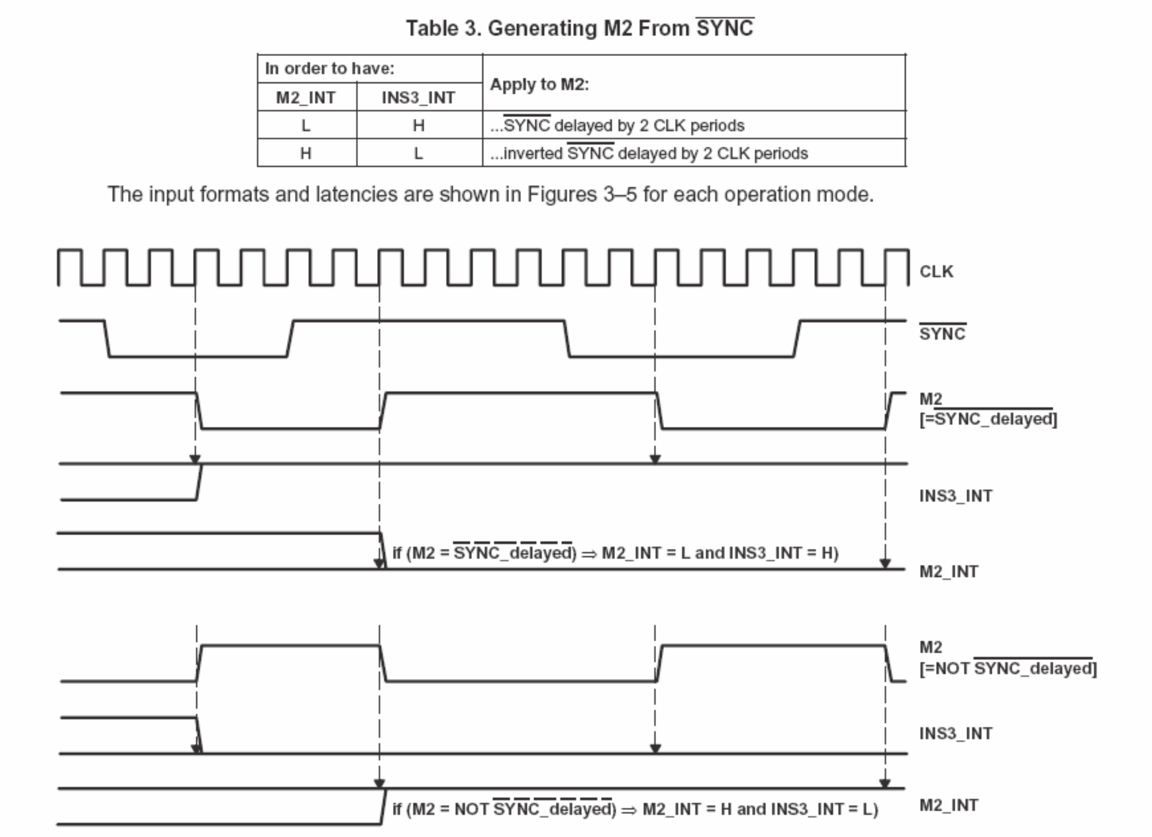


Table 3. Generating M2 From $\overline{\text{SYNC}}$

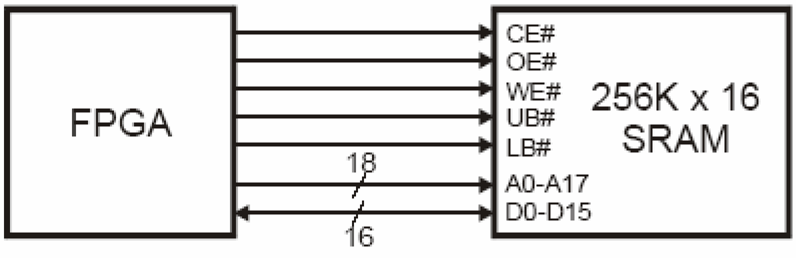
In order to have:		Apply to M2:
M2_INT	INS3_INT	
L	H	...$\overline{\text{SYNC}}$ delayed by 2 CLK periods
H	L	...inverted $\overline{\text{SYNC}}$ delayed by 2 CLK periods

The input formats and latencies are shown in Figures 3–5 for each operation mode.

CLK
$\overline{\text{SYNC}}$
M2
[=$\overline{\text{SYNC\_delayed}}$]
INS3_INT
if (M2 = $\overline{\text{SYNC\_delayed}}$) ⇒ M2_INT = L and INS3_INT = H)
M2_INT
M2
[=NOT $\overline{\text{SYNC\_delayed}}$]
INS3_INT
if (M2 = NOT $\overline{\text{SYNC\_delayed}}$) ⇒ M2_INT = H and INS3_INT = L)
M2_INT

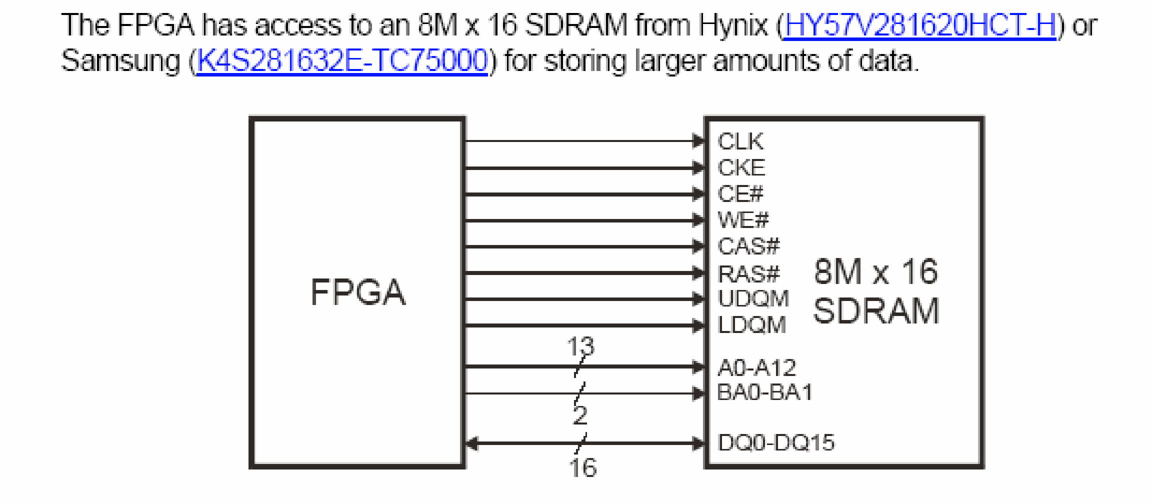
As in lab 2, we can use memory and video access to SRAM creating a video buffer. We calculate needing 40K of memory per sample. We will need to have at least 2 samples in memory at any one time. As a result, we calculate 80k of RAM needed to hold sample data. It is possible to increase the sample rate and by a power of 2 and use 160K of the 4M.

SRAM block diagram:



When implementing the video effects for the audio, it may require more memory. When the SRAM fill, we can use the SDRAM.

SDRAM block diagram:



The FPGA has access to an 8M x 16 SDRAM from Hynix (HY57V281620HCT-H) or Samsung (K4S281632E-TC75000) for storing larger amounts of data.