Embedded System Design

Project Design Document

Internet Radio

Yingjian Gu yg2154@columbia.edu
Qiutao Yu qy2104@columbia.edu
Chun-Chuen Li cl2222@columbia.edu
Imran Quyyum iq2101@columbia.edu

1. Introductin

We propose to implement an internet radio server using the XESS XSB-300E board. The previous 2005 project had implemented a RTP Streaming Audio over Ethernet server. We will start our design using the previous project as reference. However, we will implement new features that were not addressed before.

The purpose of this project is that we would be able to access as much as possible pheriherals on the FPGA board, and learn more from the embeded system design course. We would not focus on a real functional complex internet server software in this project.

We will implement the live broadcast radio with configurable back ground music and advertisement insertion. We will capture the audio through the audio codec chip and microphone, and playback the broadcast content for monitoring through the codec chip and speaker. We will use the RTP protocol to streaming out the audio through the ethernet controller. We will be able to insert advitisement to the broadcast under user control, and put back ground sound to the broadcast also. In order to implement the configurable features, we will use the VGA with menu display, and user input control from UART channel, so that the user input can be used to change the mode. We will provides the SRAM, SDRAM and Flash memory interface for the data storage. The SRAM is used to save the codec serial data during the capturing, also it is used as the VGA buffer. The Flash is used to save the advertisement sound data. The SDRAM is used as the buffer to record back ground sound and advertisement, and then these data will be programmed to Flash.

The following is the system diagram,

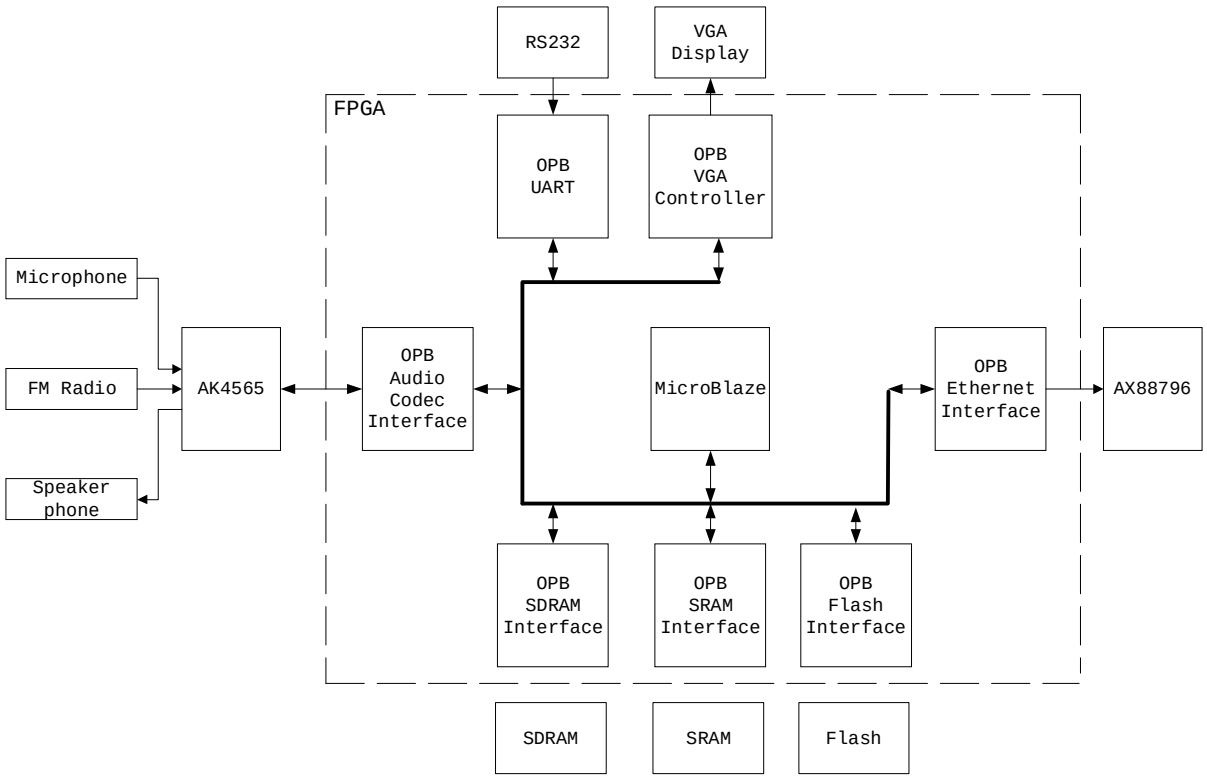


Figure 1 System Diagram

2. Implementation

2.1 Memory Interface

The on board SRAM, SDRAM and Flash will be used by the application code. The FPGA will provides interface to these memories.

The SRAM interface will handle the transfer from 32 bit OPB data bus to 16 bit SRAM data bus. All the 512KB memory space will be accessible from the processor.

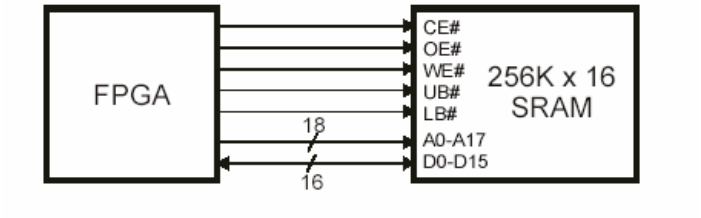


Figure 2 SRAM/FPGA connection

The Flash data bus is also 16 bit, and similar interface will be used to connect the data bus. The Flash programming and erasing will require special address pattern to initiate the command. So the FPGA will provide a Flash command register for this purpose. The special address 0x5555, 0x2AAA will be generated accordingly instead of the OPB address in this case.

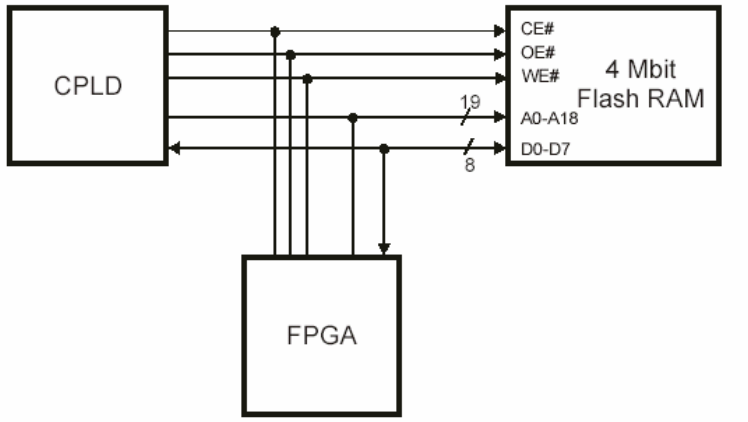


Figure 3 Flash/FPGA Connection

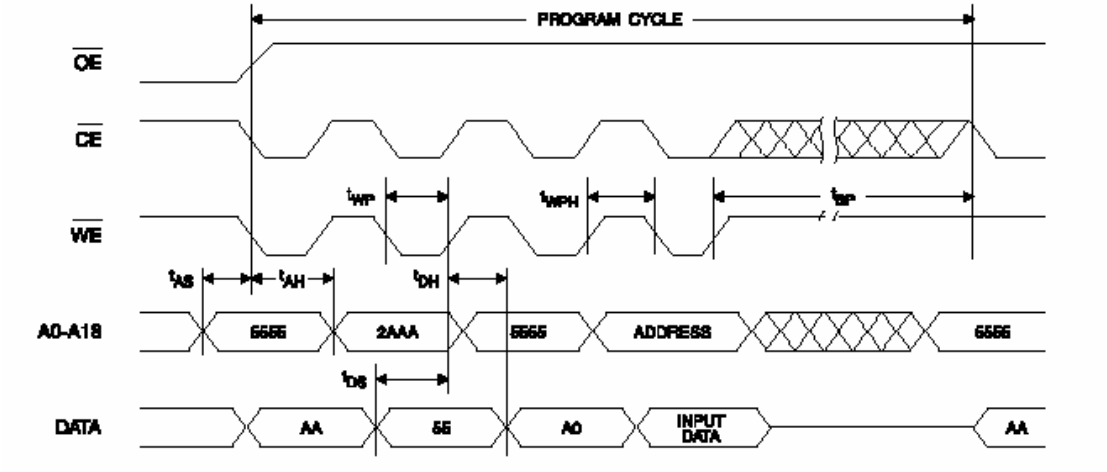


Figure 4 Flash Programming Timing

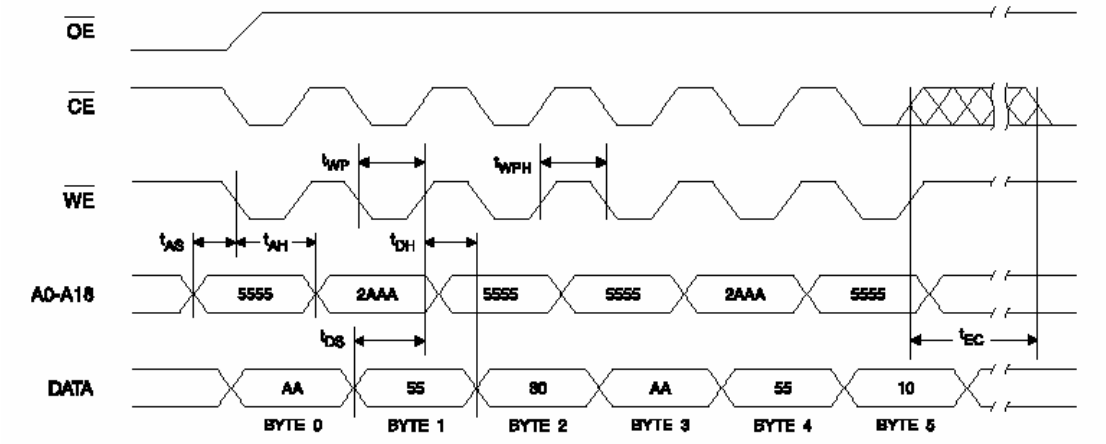


Figure 5 Flash Erasing Timing

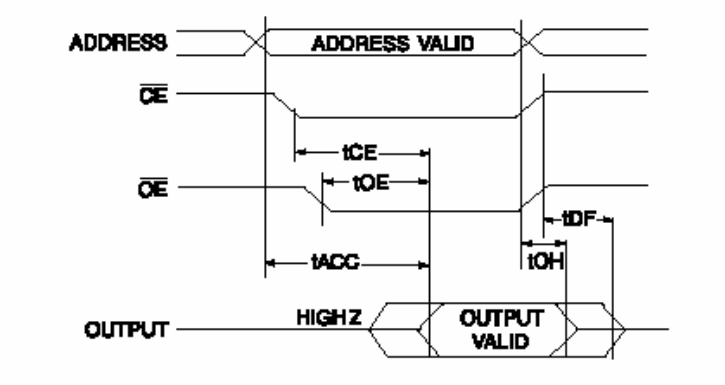


Figure 6 Flash Read Timing

The SDRAM interface will be designed such that the SDRAM is accessed in a similar way as the SRAM. The refresh operation of the SDRAM will be done by the SDRAM controller itself. The SDRAM controller provides simple read and write operation to the OPB bus.

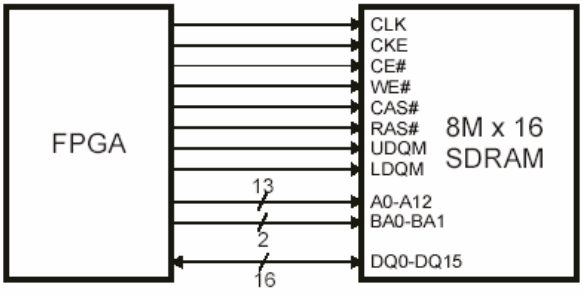


Figure 7 SDRAM/FPGA connection

We will start from the XESS SDRAM controller reference design an-071205-xsbsdramcntl.pdf. We will implement our own logic with OPB interface. The following is the SDRAM timing diagram from this application note,

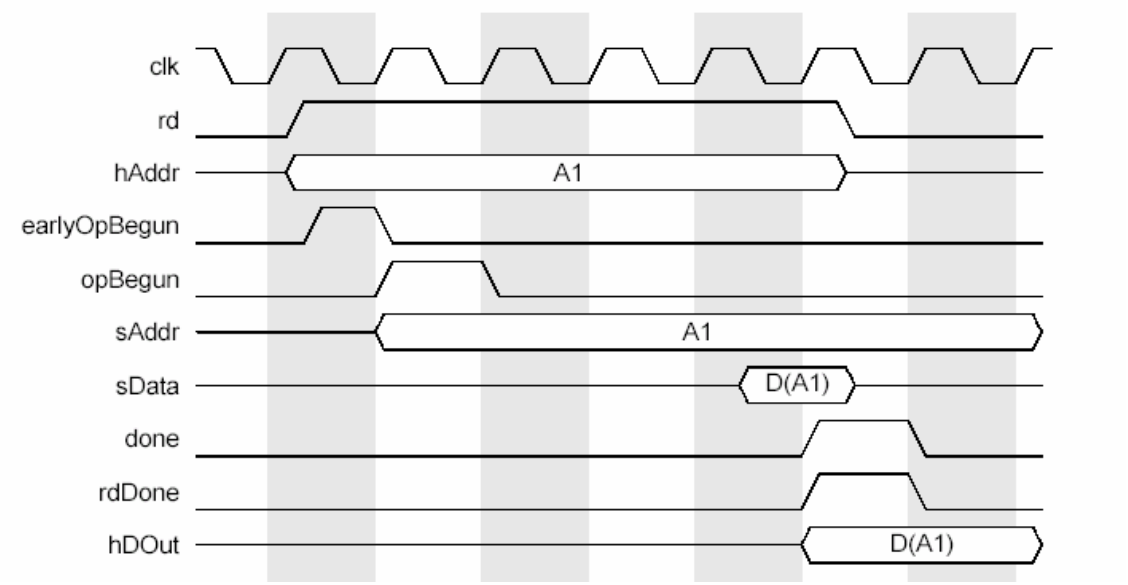


Figure 8 SDRAM Read Timing

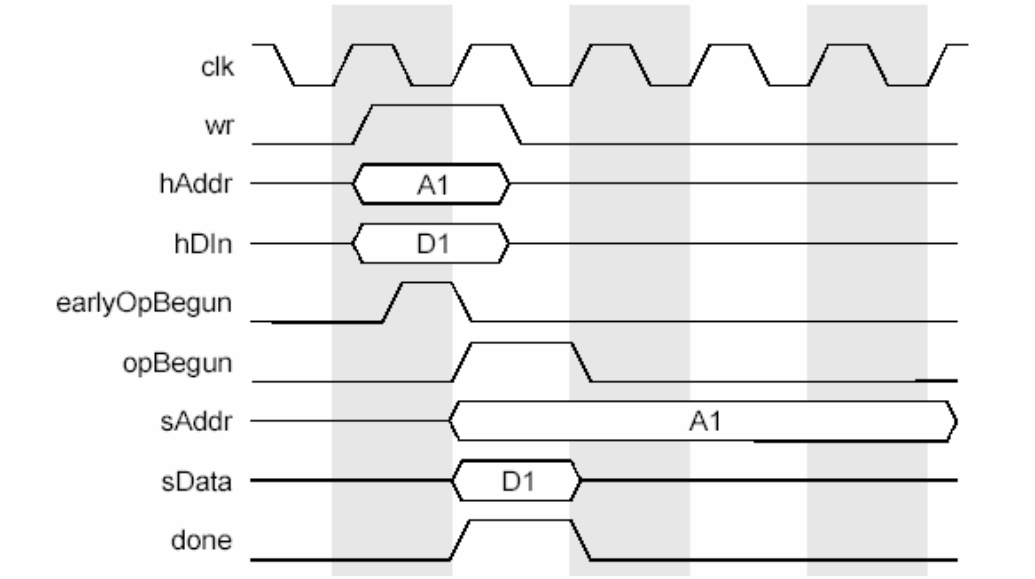
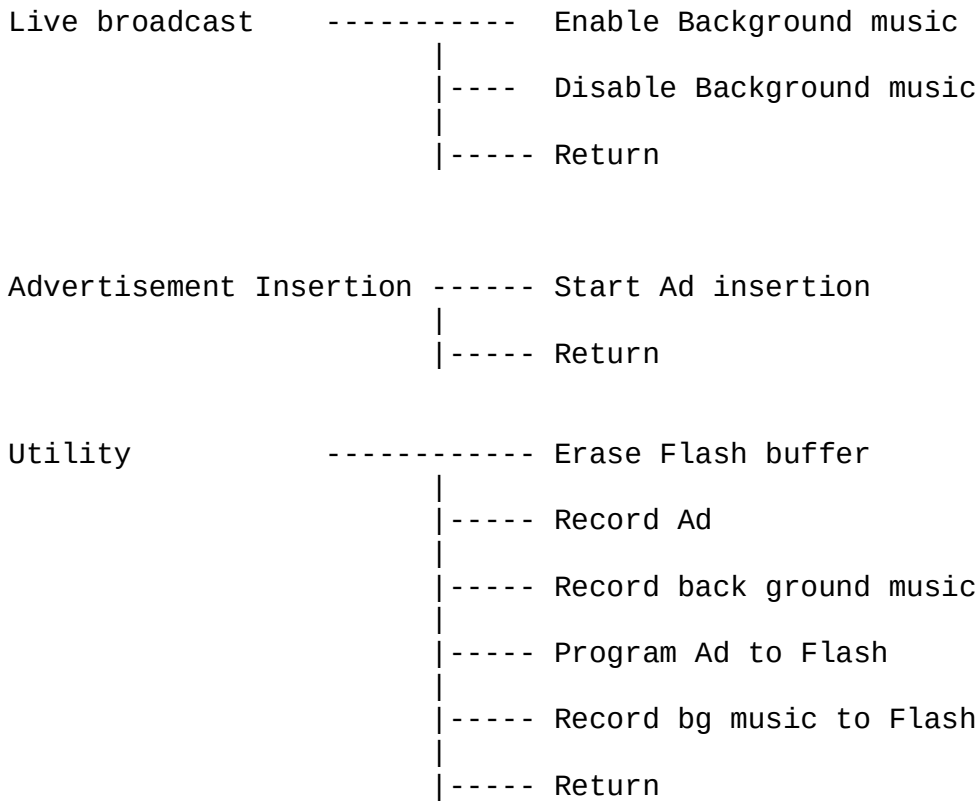


Figure 9 SDRAM Write Timing

2.2 User Interface

The VGA display and UART is used to implement user interface. The operation of the server is controlled by the menu displayed on the VGA. The user select the menu command from the UART channel. Based on the user inputs, the operation of the server can be changed, such as switch between live broadcast and advertisement, enable background music, etc.

The following is the menu configuration:



We will use fixed address in the Flash to save the advertisement and back ground music. The size of the data is also fixed.

We will use the text mode VGA first. If time permits, we will implement the graphics mode VGA for better visual effects.

The menu selection will be controlled from the UART channel, and the menu font should be highlighted when it is selected.

2.3 Live Broadcast

The audio input is taken from the AK4565 Audio Codec. The Microphone is used to input sound for the broadcast. The audio codec interface is built in the FPGA, and serialized bit stream is saved in the SRAM buffer.

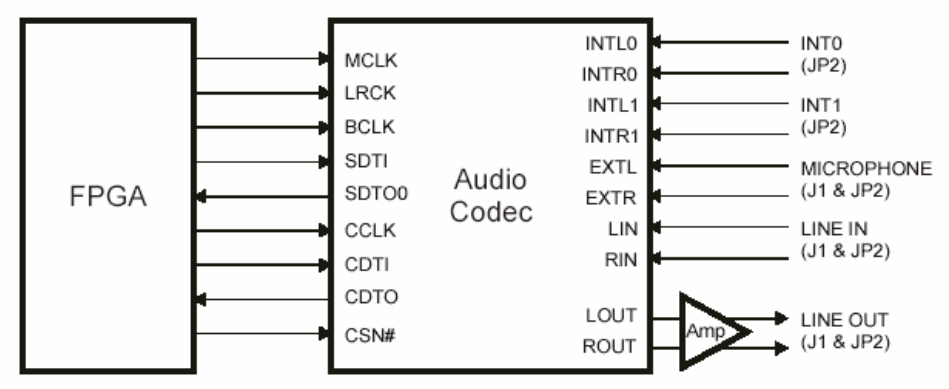


Figure 10 FPGA/Audio Codec Connection

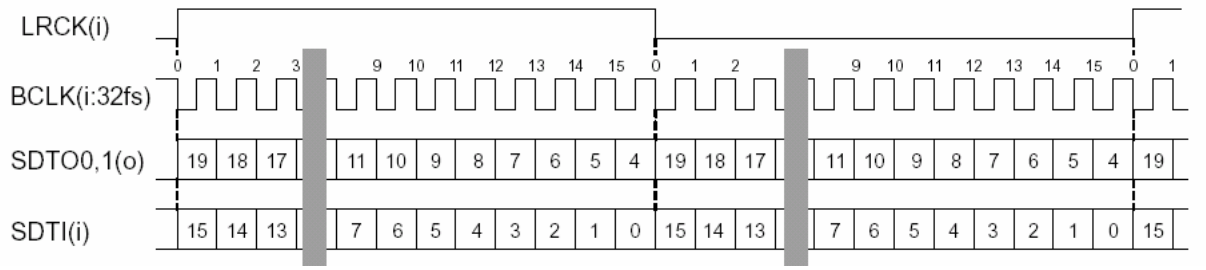


Figure 11 Codec Timing Diagram

2.4 Advertisement Insertion

The advertisement clip would be inserted to the broadcast. The source data for the advertisement is saved in the Flash memory in advance, and sent to the ethernet controller when the advertisement mode is enabled.
The Flash OPB interface need to be designed in the FPGA. The 16 bit Flash data bus to 32 bit processor data bus conversion will be done automatically. The Flash programming and erasing command will be started by the software, as well as the polling of the status.

2.5 RTP Streaming

The RTP protocol will be used for the audio streaming to the internet. The processor will take the payload data from the buffer, and construct the IP/UDP/RTP packet. The Ethernet Controller ASIX AX88796L will be initialized and used to transimit the audio data to the internet.

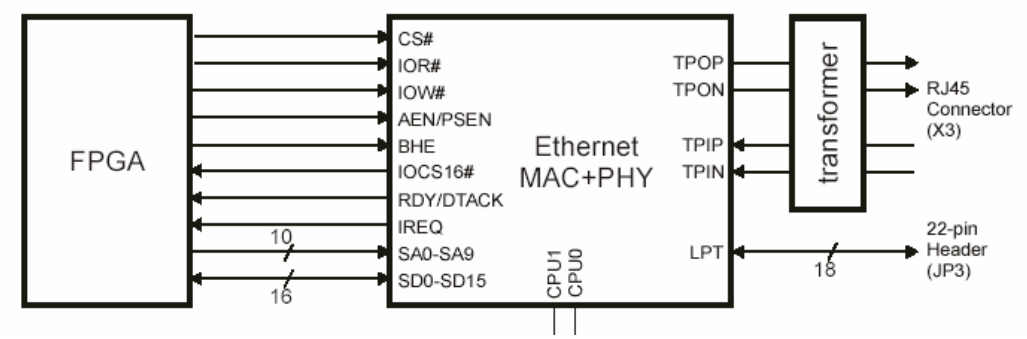


Figure 12 FPGA/Ethenet Connection

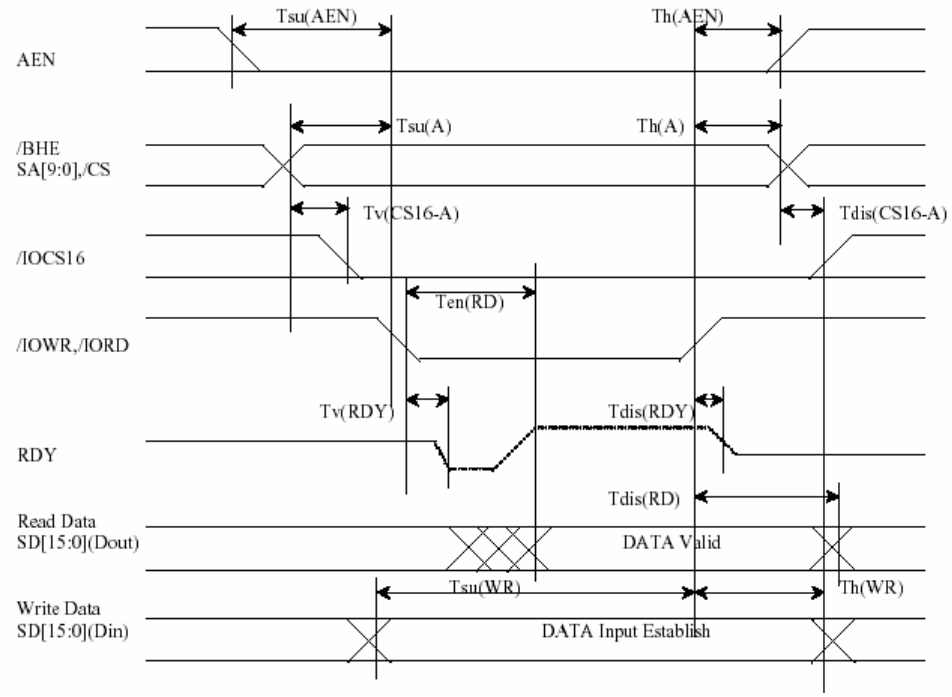


Figure 13 Ethernet Timing

2.6 Background Sound Mix

The background sound is saved in the Flash memory in advance. The source data is fetched and mixed with codec input data, then saved to the ethernet output buffer for transimision.
The mix will be done by the software. When the end of the backgroud sound buffer is reached, the code will loop back to the beginning of the buffer again.

2.7 Audio Loop Back Monitoring

The broadcase contents will be converted back into an analog output signal by the FPGA and AK4565, and played out from the speaker phone connected to the board line out. Then it is possible to monitor the broadcast on the internet in real time.

2.8 Ad/Background Sound Recording

In order to facilitate the advertisement and background music feature, the line in will be sampled in the same way as the broadcasting, and the serialized codec data will be saved in the SDRAM memory temporarily.
The size of the captured data will be in fixed length to simplify the design. The SDRAM buffer is then programmed to the Flash memory, so that the contents are not lost after power down.

2.9 PCM/MP3 Streaming

The PCM audio will be the default format for the streaming. If time premits, we will try the MP3 streaming mode. The MP3 steaming will require a MP3 encoder. It is not possible to develop MP3 encoder from scratch, and we intend to use existing MP3 decoder source code from the Lame tool. It is available under GNU GPL license.
http://lame.sourceforge.net/