# Audio Conference Communication System (ACCS)

## *Design Proposal*

**Team Members:**

Xianzhen Zhu (King)

Saher Ahwal

Wegene Tadele

# 1. Abstract

As digital system engineers, we plan to design and implement an audio digital communication system for group conferencing. Whether you are an employee in a big company or a small one, individuals and groups in your building need to meet and communicate to make plans and work on milestones. We propose a design of a wired conference communication system which enables short meetings and calls between individuals and teams in the company with a single communication channel. A certain number of bases are to be installed in different regions, and each connected to a keypad, a headset and an FPGA. The system has a central coordinator that directs and manages communication among node stations. Each base will transmit the audio/video signals to the coordinator. Then the coordinator routes/forwards the call to the corresponding node station(s) depending on the identification address(s) sent by the caller.


# 2. Introduction and Overview

Digital communication has been advancing rapidly in the recent years, giving rise to a great deal of communication devices (e.g smartphones) and communication software (e.g Skype). Whether we are talking about cellphones or video communication software, there is dependency on the cellphone network and the internet connection respectively. For workplaces that rely heavily on audio and video communication between teams, there is a need for a more robust communication system. Therefore, we are planning to design and implement a digital communication system that allows people within a certain workplace to initiate audio conference without relying on unpredictable network failures and limited bandwidth of cellphone networks.

The system will have a coordinator base which will also act as a central phone station. A certain number of bases will be installed in different regions and are going to be connected to the coordinator base. The system will allow node stations (bases) to call to more than one node stations concurrently. If no one was available at a called base or the base was busy in a call, the caller will have an option to record a voicemail. A caller at the base will be able to listen to his voicemail and delete chosen messages from memory as well. A participant in a group call has the option to add a new person to the same call. If one tries to call a busy base (in active call), the busy tone should be heard by the caller.

The project will be divided into multiple phases. In the first phase, the main features of voice communication are implemented like: having an audio conference conversation, voicemail and simple display of ongoing calls. In the next phase, we will implement extra features like: incorporating multiple ringtones, adding a busy tone, and using phone pads to initiate calls and listen to voicemail.

The addition of video communication scheme can be considered as a stretch goal. If time permits, we will add video features to our communication system. This will allow users to initiate video conference calls and record video mail in other users' machines.

# 3. Implementation

The following is a detailed description of the implementation of the Audio-Video Conference System (AVCS). First, we will present the block diagram of the system including the modules. Next, we will go through the detailed description of the modules.

## 3.1. Block Diagram

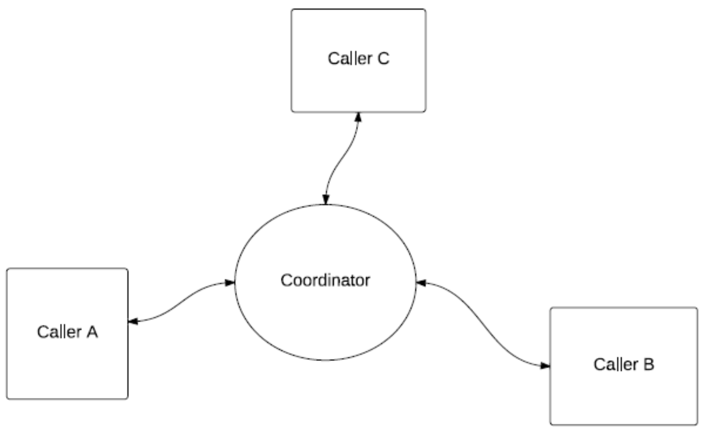


Figure 1-Three Users Based Station

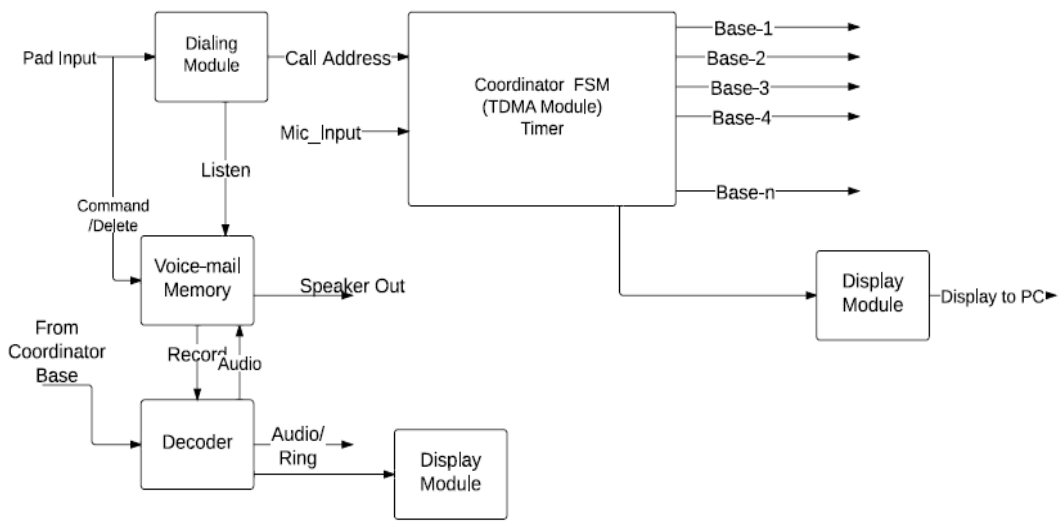


Figure 2-Higher Level Block Diagram

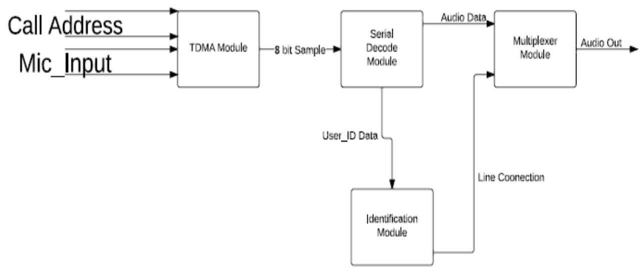


**Figure 3-Coordinator FSM Block Diagram**

## 3.2. Module Descriptions

### 3.2.1. <u>User End Module:</u>

### 3.2.2.

- **Dialing Module: (*Wegene*)**

  This module is implemented on all stations and is responsible for decoding the dial/call. It takes the input possibly from lab kit switch or external keypad as calling a specific user end and transmits the information to the coordinator station.

- **Voicemail Module*(Saher)***

  This module is responsible for recording and listening to voicemail. The inputs determine whether the user wants to listen to voicemail, or delete/clear his voicemail. Furthermore, this module is responsible for saving the audio voicemail received after a certain timeout is passed.

- **Decoder Module (*King*)**

  This module decodes the received communication signal from the coordinator. It determines whether the incoming signal corresponds to a call which triggers the ring tone, or audio data which outputs to the speakers or triggers voicemail recording.

- **Display Module(Saher)**

  This module is responsible for displaying a simple user interface on the screen at each station. This includes a display of incoming call information in addition to some simple graphics.

### 3.2.3. <u>Coordinator Station Modules:</u>

The following modules are implemented only on the coordinator station which is, in general, responsible for controlling the conferences by time division multiplexing communication protocol, the bookkeeping of ongoing calls on the FPGA memory, and the outgoing of voicemail signals.

- **Coordinator FSM Module(all)*:***

  This module set up windows for different end users. Every period each user end would have two windows for communication with base station. The first window is used for the information of who the user end is calling. The other window is utilized for transmitting information of audio inputs. This is essentially using a TDMA to access a single channel for multiple conversations.

- **Display Module (Saher)**

  This module is responsible for displaying information about the whole system status at the coordinator station. This shows a graphical interface of the stations and the ongoing calls. It shows occurring conferences and busy stations

- **User identification Module: (king)**

  This module promises a user end make correct connection to another specific user end. It includes data of each station's identification numbers. It is essentially informing the address of the caller

- **Serial data decoder Module(King)**

  This module enables the system to decode data from serial data transmission. The ideal situation is every eight clock cycles the module would sample a data from the last eight bits.

- **Multiplexer Module (Wegene)**

  This module allows the output to reach the desired end users. When one line is connected, audio data only transmits to the connected stations. The decoder module receives a single bit input from the serial data decoder module and outputs a byte (8 bit) data.