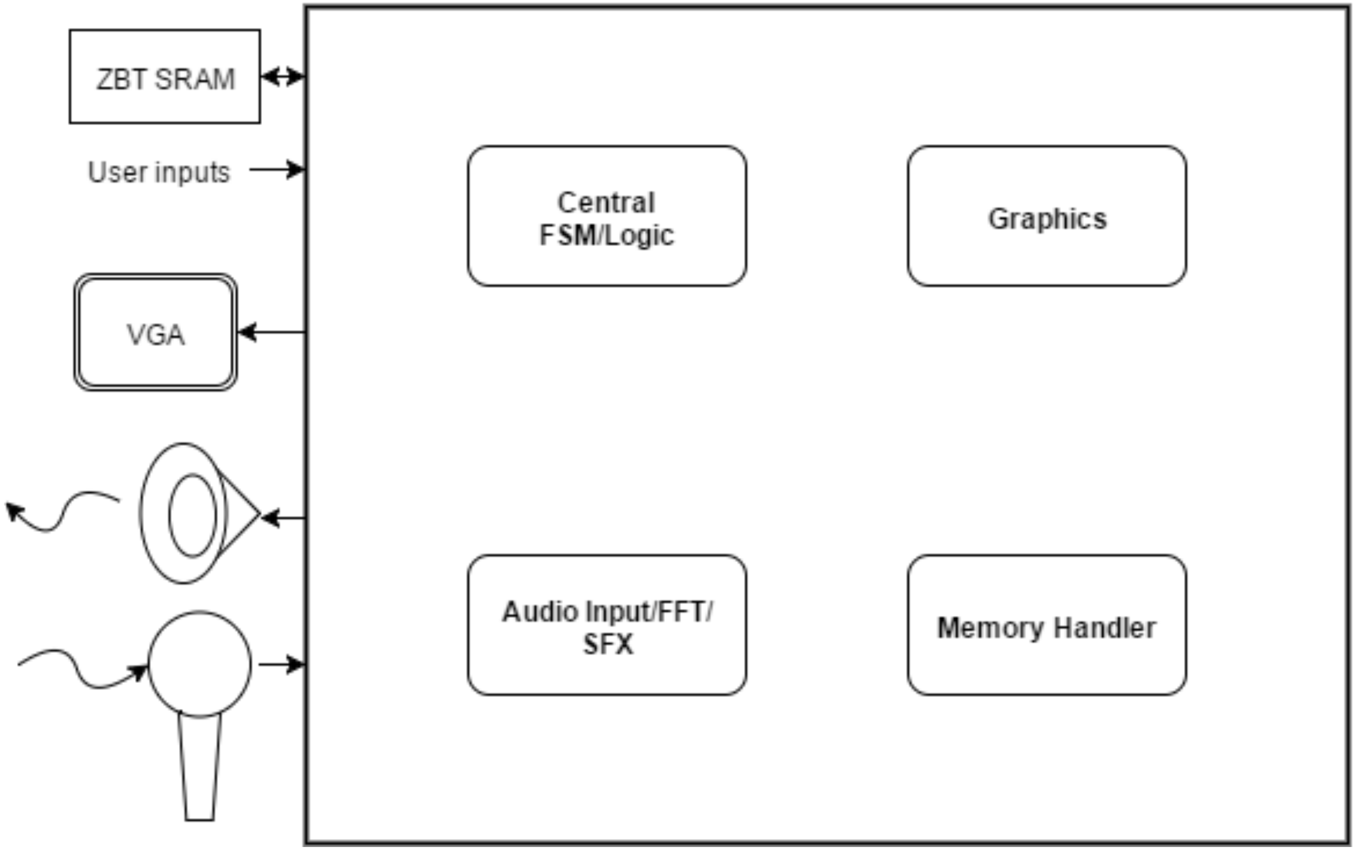
Real-Time Sound Analysis / Synthesis

Michelle Qiu
Germain Martinez
Gerzain Mata

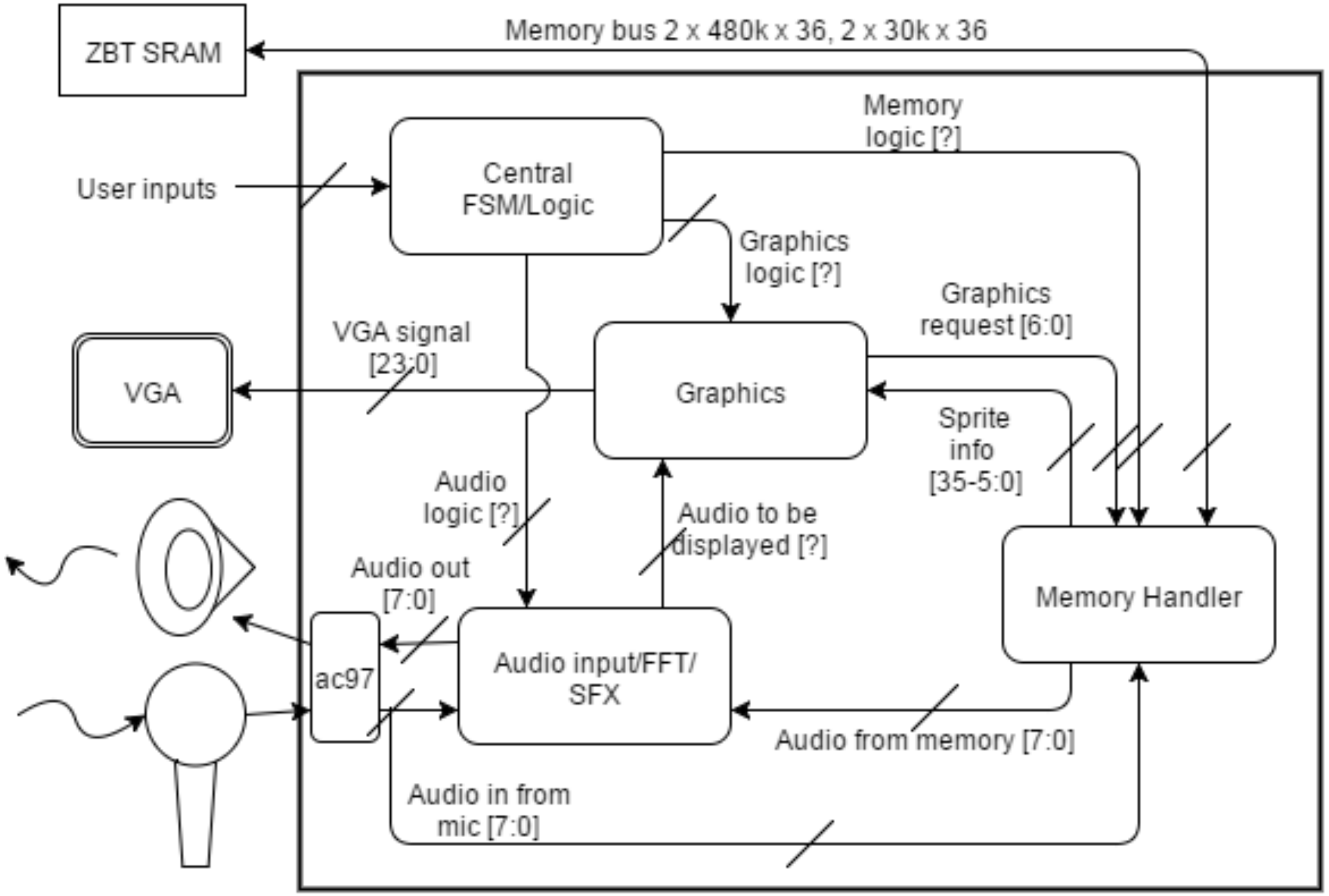
Overview

- Use FPGA buttons and switches for navigation
- The user sings/ hums/ plays an instrument
- A microphone picks up the sound
- The sound is saved onto memory for later playback
- Audio effects can be added to the sound samples
- Multiple audio samples can be recorded and played

Main Modules and External Components



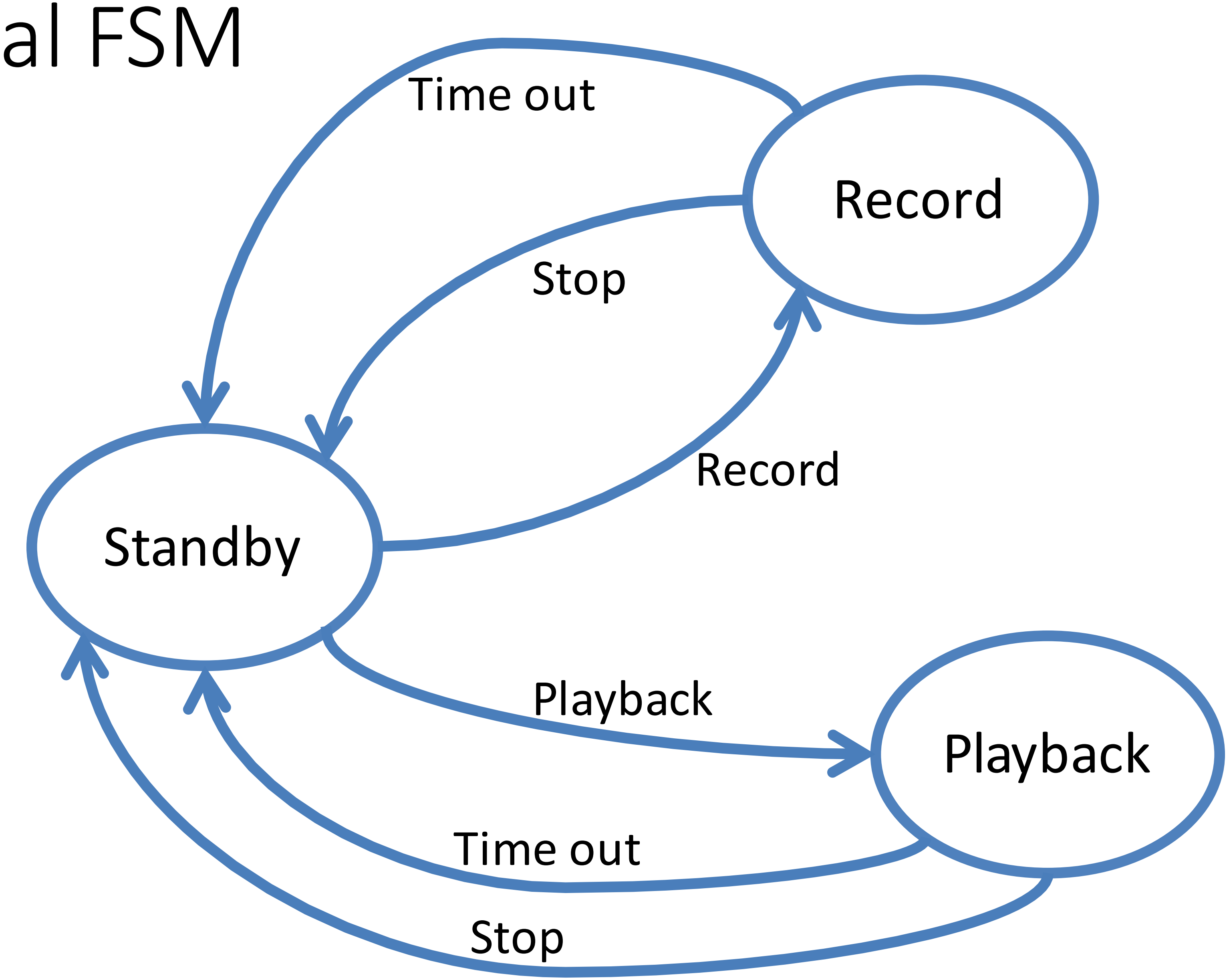
Block Diagram



Specific Number Figures

- Use ZBT Memory – two 512k x 36 bit memory bus
- 24 kHz sampling rate and 12 bit encoding scheme
- 288 kilobits per second data rate
- ~80 kAddresses of ZBT memory required for 10-second recording
- Use one bank of memory for each sound recording
- 2952K bits of BRAM available for images for graphics

Central FSM

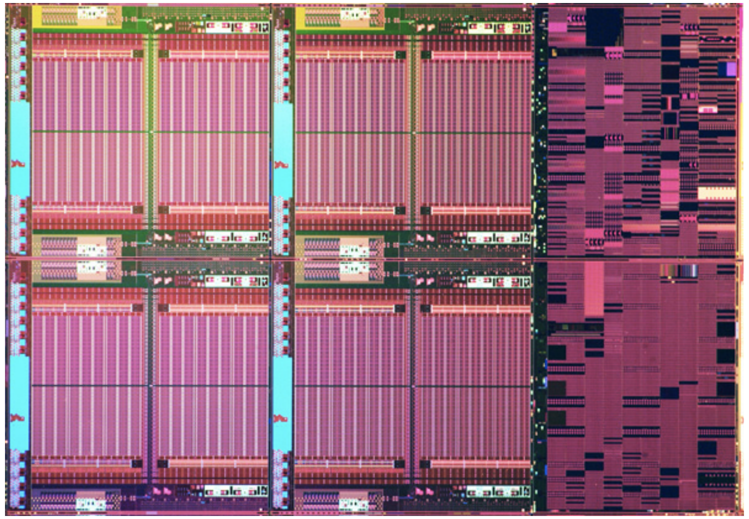

Time out
Record
Stop
Record
Standby
Playback
Playback
Time out
Stop

Central FSM

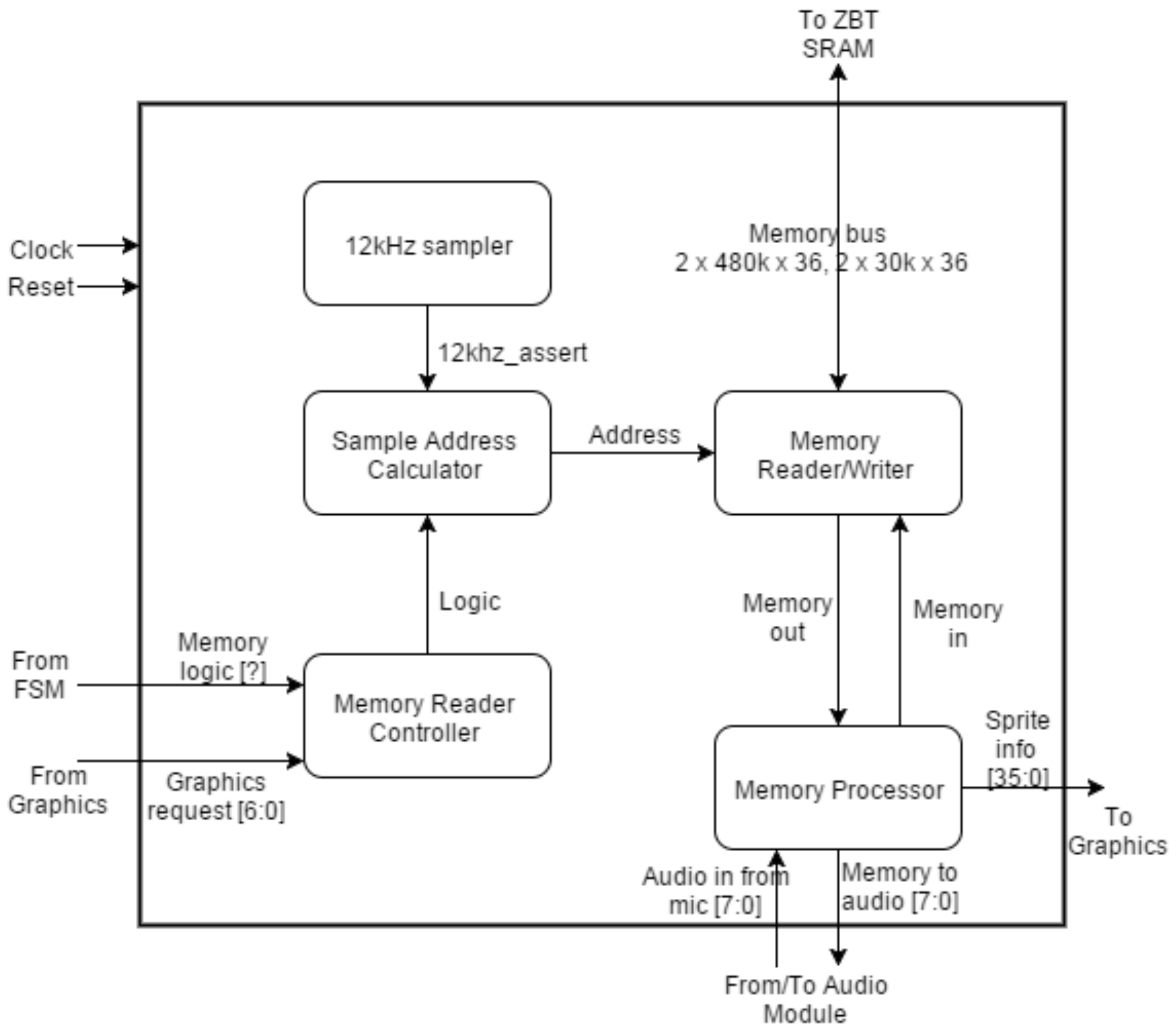
- Processes user inputs
- Coordinates the modules
- Choose which bank to record to and what effects to apply when playing back
- Each team member implements part of the FSM most pertinent to their individual module

Memory Handler

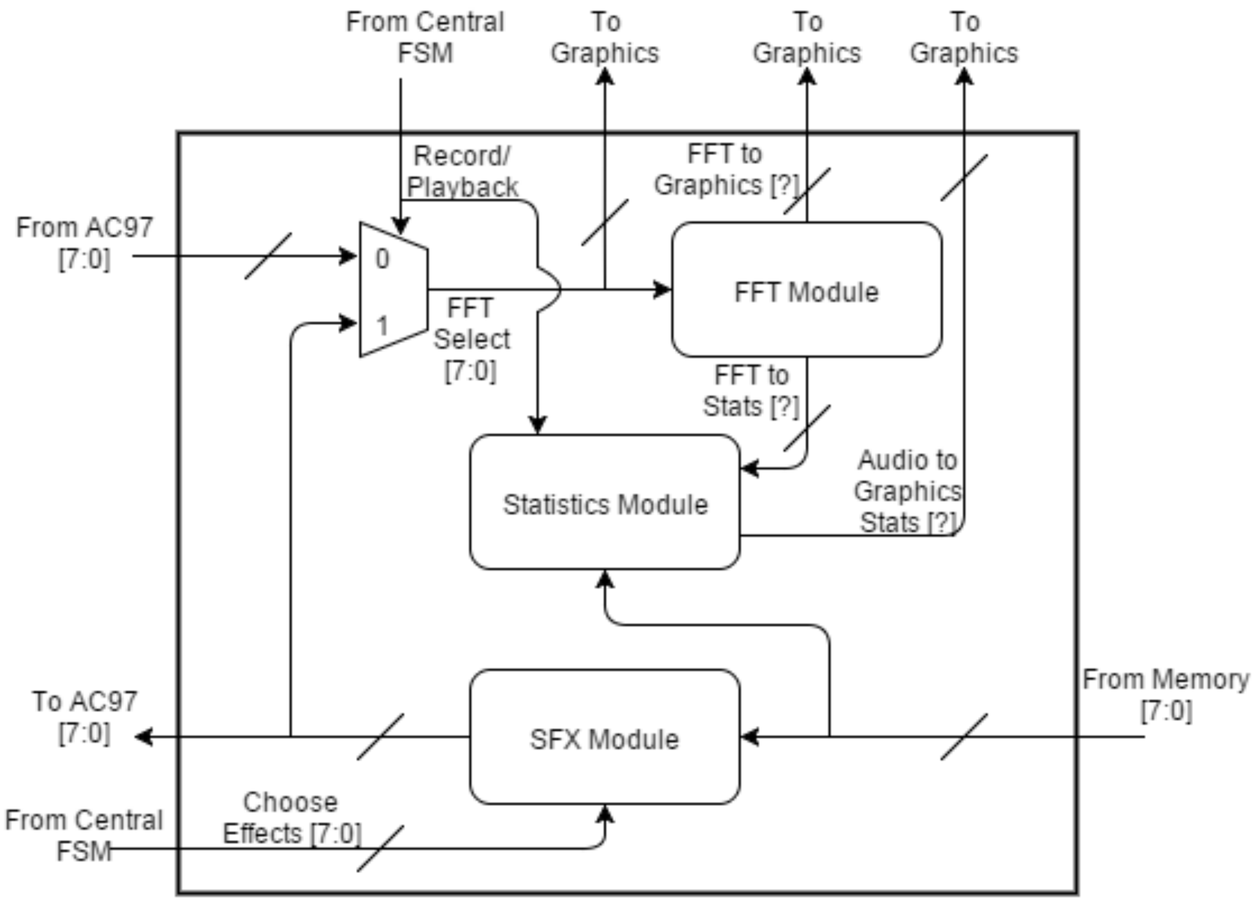
- Take in input from microphone
- Write audio samples to memory
- Retrieve sound samples from memory upon request



Memory Handler – Block Diagram



Audio Module - Block Diagram



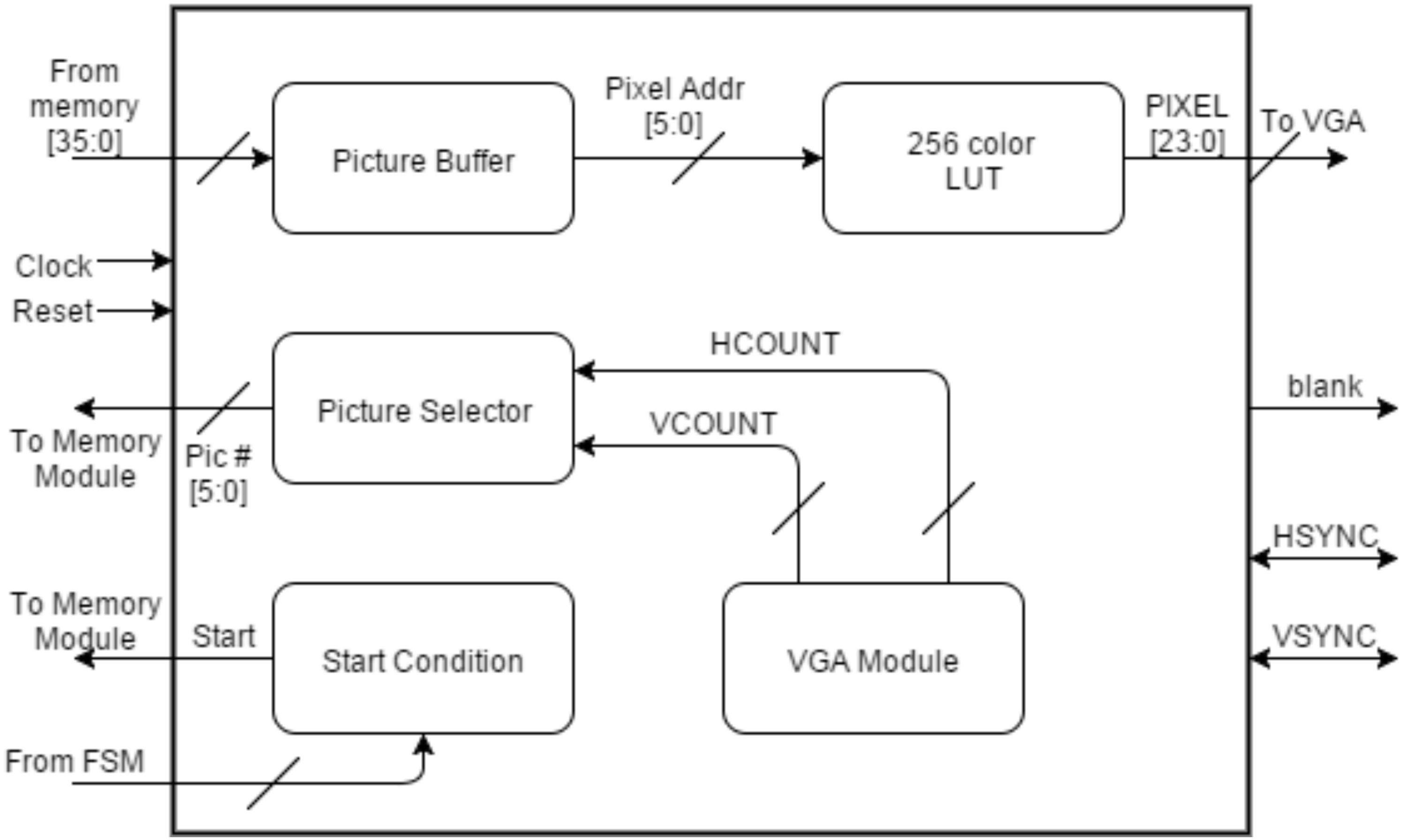
Audio Module

- Process input from microphone and write to memory
- Take effects from FSM and apply them to recorded audio
- Output FFT data to Graphics module
- Output sound to speaker
- Most complex part of project

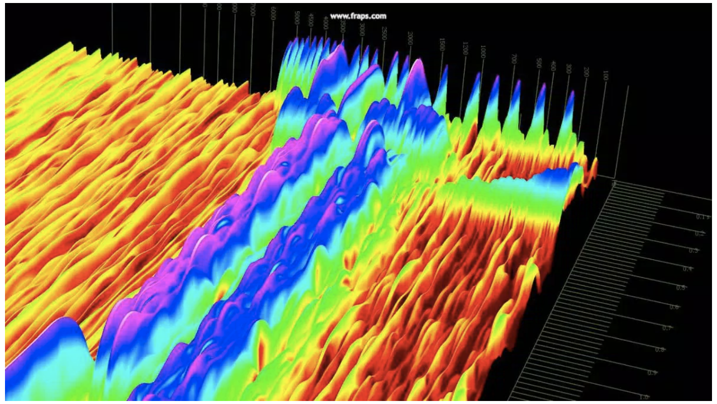
Audio Module - Proposed Effects

- Compression & Limiting
- Delay & Echo
- Expansion & Noise Gate
- Phasing
- Chorus (if time allows)
- Vocoders (stretch goal)

Graphics Module – Block Diagram

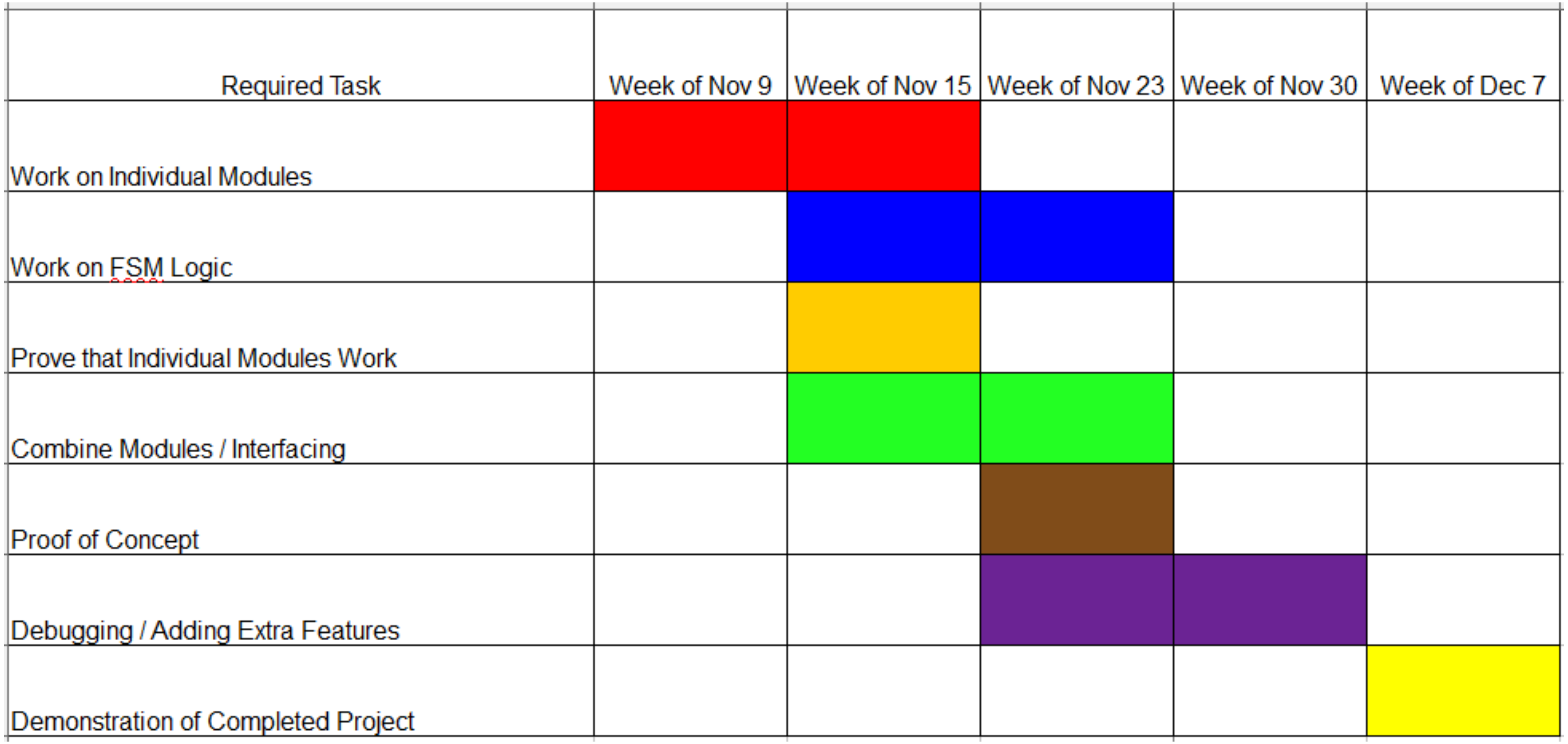


Graphics Module



- Take in graphics from memory
- Output FFT spectral data
- Display running statistics (predominant frequency, amplitude, sample characteristics)
- Take in preloaded data from block ROM
- Data saved as 6 bit pixel encoding
- 6 bit pixel encoding -> 24 bit color to VGA -> LUT
- Minimum 30+ different images for text/numbers/etc. (30 kBits)

Task Timeline



Required Task	Week of Nov 9	Week of Nov 15	Week of Nov 23	Week of Nov 30	Week of Dec 7
Work on Individual Modules	🟥	🟥			
Work on FSM Logic		🟦	🟦		
Prove that Individual Modules Work		🟧			
Combine Modules / Interfacing		🟩	🟩		
Proof of Concept			🟫		
Debugging / Adding Extra Features			🟪	🟪	
Demonstration of Completed Project					🟨

Forseeable Challenges

- Getting FFT and statistics in real-time
- Simultaneous and offset playback of different recordings for sound effects and playback
- Effect complexity
  - Possible source of latency
  - Multiple effects at the same time?
- Playing multiple sound clips at the same time