Concept Description

This project implements a **basic digital audio synthesizer** using an FPGA(Nexys A7 FPGA board with Artix-7). It generates square wave audio tones through **PWM (Pulse Width Modulation)** based on input from **16 slide switches**. Each switch corresponds to a different audio frequency. When a switch is ON, the FPGA calculates a PWM limit to control the output waveform's frequency. The result is output through AUD_PWM (audio signal) and AUD_SD (audio enable).

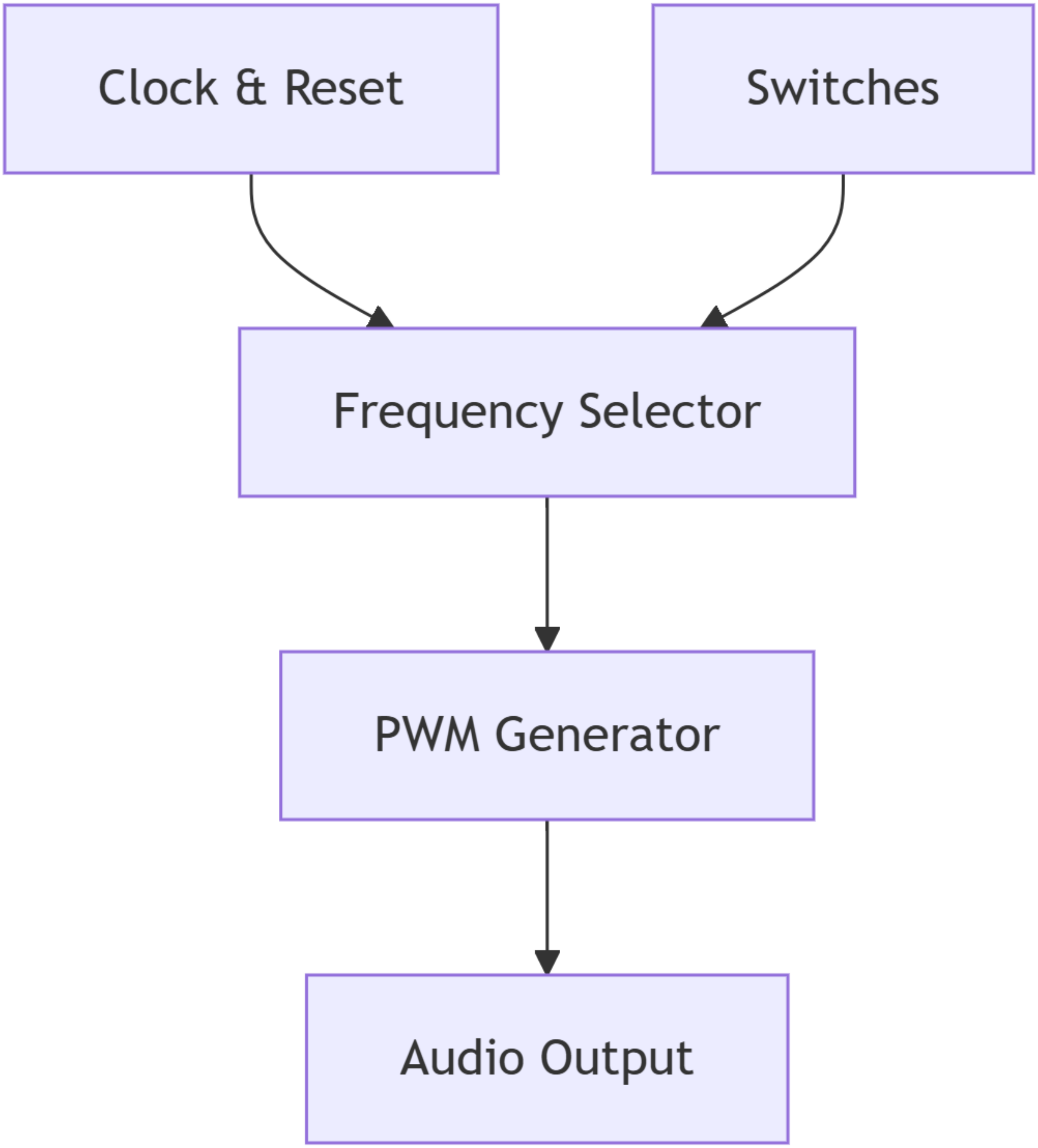
System Requirements

- Nexys A7 FPGA board with Artix-7
- 16 slide switches (SW[0] to SW[15]) to select tone frequencies
- Audio output through 3.5mm jack (AUD_PWM and AUD_SD)
- 100 MHz input clock
- Reset button to mute output and reset PWM counter

Used Tools

- **Vivado**: For writing VHDL code, simulation, synthesis, and FPGA programming
- **Nexys A7 Board**: Artix-7 based FPGA development board
- **KiCad** was used for schematic capture and PCB layout. It helped convert the working VHDL design into a minimal standalone hardware version by designing a custom PCB.
- **Github:** Additionally, I used **Git** for project version control and **Kanban** (in GitHub Projects) to manage tasks and track progress efficiently.

Block Diagram:



Used Peripherals

- **Slide Switches** (SW[0] to SW[15]): To select tone
- **Audio Output Port** (AUD_PWM, AUD_SD): Sends the tone to speakers

- **Reset Button**: Resets the system
- **On-board Clock**: 100 MHz system clock
