Verilog Projects

1. Arithmetic Logic Unit (ALU)

Design an ALU capable of performing core operations such as addition, subtraction, and bit wise functions, typically used in processors.

2. Finite State Machine (FSM) Design

Build a simple FSM for a practical system, like a washing machine, managing states and transitions based on inputs.

3. Seven-Segment Display Controller

Build a module that translates binary inputs into signals for a seven-segment display, allowing numerical digits to be shown.

4. Traffic Light Controller

Develop a system to control traffic lights at an intersection, sequencing lights based on preset timers or traffic density.

5. UART Communication Module

Design a module that enables serial communication by converting parallel data into serial form for transmission and vice versa for reception.

6. PWM Signal Generator

Implement a circuit that generates a PWM signal, where the width of the pulse can be adjusted to control devices like motors or LEDs.

7. SPI Controller

Create an SPI interface that coordinates data exchange between a master device and one or more slave devices using clock synchronization.

8. Integer Division Unit

Implement a circuit to divide integers, managing both quotient and remainder calculations for signed and unsigned numbers.

9. Booth Multiplier

Develop a multiplier circuit using Booth's algorithm to handle both positive and negative numbers more efficiently than standard methods.

10. Barrel Shifter

Create a hardware shifter that can rotate or shift data left or right by a specified number of positions instantly.

11. AXI-to-APB Bridge

Develop a bridge that converts high-speed AXI protocol signals to the simpler, slower APB protocol for peripheral device communication.

12. Integer Division Unit

Implement a circuit to divide integers, managing both quotient and remainder calculations for signed and unsigned numbers.

13. Dual-Port RAM

Design a memory block with two separate access ports, allowing read and write operations to happen simultaneously without conflicts.

14. Basic Processor Design

Construct a simple processor that can interpret a set of basic instructions to perform computational tasks.

15. Pipelined Processor

Enhance a processor design by adding pipeline stages, allowing multiple instructions to be processed in different stages simultaneously.

16. DSP (Digital Signal Processor) Design

Create a processor optimized for executing complex mathematical functions used in digital signal processing, like FFTs or convolutions.

17. Digital Filter Implementation

Build a hardware filter that processes digital signals to remove noise or extract certain frequencies, like an audio equalizer.

18. Ethernet MAC Controller

Design a controller that handles the sending and receiving of Ethernet frames, managing data flow at the data link layer.