ECE 337: ASIC Design
Lab 8

APB-UART: Advanced Peripheral Bus Slave-attached UART Rx

Week 8

Deadlines & Reminders

- Lab 8 Deadlines
  - Required Preparation Phase (Design planning)
    - Due by Late Night two days before your lab 9 (Week 9)
    - Example: Late Sunday Night for Tuesday Lab Sections
  - Required Verification Sign-offs
    - Due by start of your lab 9 (Week 9)
  - Automated Design Grading
    - Due by start of your lab 9 (Week 9)
- Submission Notes
  - Must achieve 50% on latest mapped grading to pass the lab
  - Top level port names and module names must be IDENTICAL to what is listed in the manual or your submission attempt will be wasted.
- Check prior scores & attempts used
  - Submit Lab8 -c

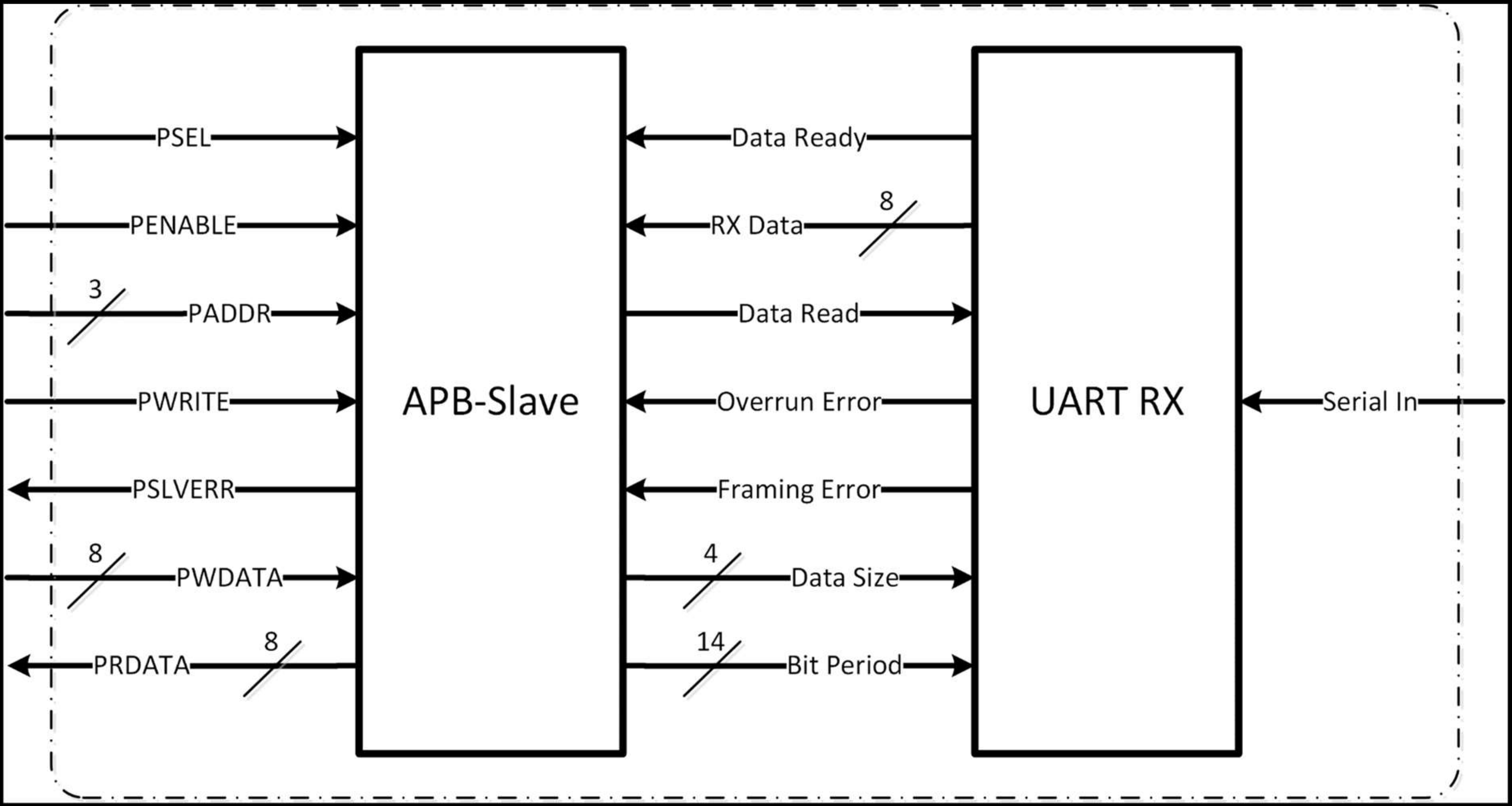
APB Highlights

- Part of ARM's AMBA protocol suite
- Low-performance bus for slow peripherals (like UART)
- 2-cycle minimum latency for transactions
  - 1st cycle for supplying/decoding address and request mode
  - 2nd cycle for response (This may be stretched by slave designs)
- Non-pipelined -> bus is fully reserved by each transaction -> any stalls mean whole APB bus stalls

In lab

- Completing the design of the APB-Slave interface
  - Provided with APB-slave interface bus model
  - Provided with starter TB (will need to add test cases)
- Update your UART Rx block
  - Update to allow configurable size of data field within packet
  - Update to allow configurable length of packet bits
- Connect two modules together for full design
  - Blend the test benches and test cases from APB and UART module
    - APB-slave configuration testing
    - APB-slave data/status read testing
    - UART data reception testing

Design Architecture



Non-used optional APB Signals

- No need for design driven ‘pready’
  - Using polling of data status register instead, to wait for data
  - Always a constant ‘1’ during transactions with this design
  - Can be statically set during connection to switching fabric
- No need for ‘pstrobe’
  - All transactions are already expected in 1-byte steps
  - Switching fabric can handle ‘pstrobe’ signal handling
- No need for ‘pprot’
  - No security variations between slave held values

    -> constant value for all transactions to this design
  - Can be handled by switching fabric

Sequence of Operation

1. Configure UART settings via writes to APB-slave config registers
2. Periodically poll the data and error status registers in the APB-slave
3. Once new data has been successfully received
   -> read from data buffer via APB-slave
4. Repeat steps 2-4, until different configuration is needed

Things you should check:

- Have you checked for non-nominal bit periods?
  - Including under different settings for nominal bit periods
- Have you check that read-only sections of APB-Slave activate ‘pslverr’ during attempted writes?
- Have you checked that ‘data status’ stays current with state of the ‘data buffer’ for the APB interface?

- Are your makefile variables filled in