IIT JODHPUR

Major Examination (OPEN-BOOK): EEL7770 Formal Verification (Dec'23)

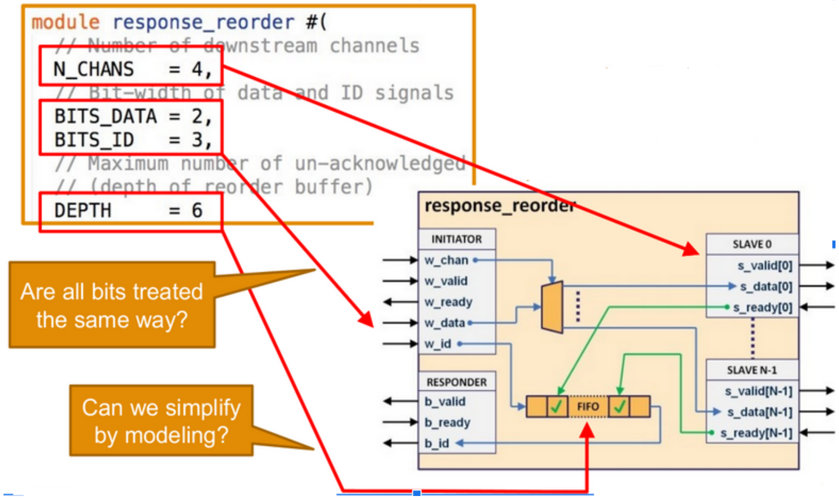
Guidelines (Total time: 120 minutes, Maximum Marks: 30):

- Please read the question paper very carefully (both sides of this paper). **NO clarification is required in any question.** In case of any doubt, assume whatever you wish to and state that in your answer.
- This is an OPEN-BOOK examination. However, usage of internet is NOT allowed.
- Please adopt mathematical reasoning as far as possible in your answers.

---

1. It is observed that a 4:1 multiplexer is not working correctly, therefore, some debug is required for this multiplexer design. It is known to you that either one of the four inputs or one of the two select pins are constantly getting connected to 1/0. Devise a mechanism using satisfiability (SAT) methodology to pinpoint (localize) this issue quickly. [4]

2. Mr. XYZ is a senior engineering technical manager in a semiconductor product company. Being a formal verification expert, he often teaches techniques for effective usage of formal methods to new college graduates joining his team. Below is one screenshot from his presentation on methods to achieve abstraction of large designs. This design is related to a communication interface between master agent and multiple slave agents. Specifically, this refers to a response reordering module that is tasked with serializing responses between the master and slaves. Because of the complexity of the actual design, we can try to simplify through some techniques for verification. [2+1+3+2]



```
module response_reorder #(
  // Number of downstream channels
  N_CHANS   = 4,
  // Bit-width of data and ID signals
  BITS_DATA = 2,
  BITS_ID   = 3,
  // Maximum number of un-acknowledged
  // (depth of reorder buffer)
  DEPTH     = 6
```

Are all bits treated the same way?

Can we simplify by modeling?

(a) Under what conditions, the number of channels (N_CHANS) can be reduced from 4 to 2 in the abstract design? What will be the advantage in terms of formal verification efforts?

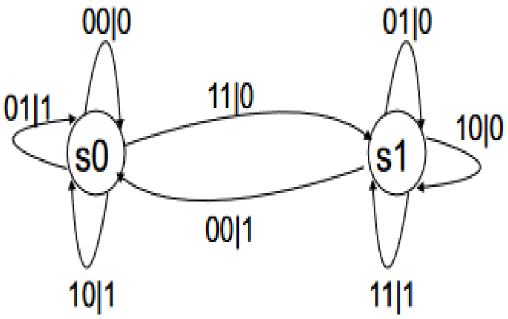
(b) What will be the impact of changing Bit-widths of data and ID signals on FIFO depth in the abstract version of the original design?

(c) FIFO stores some un-acknowledged responses from slaves and then serializes them for providing to the responder. Write property/properties for checking this.

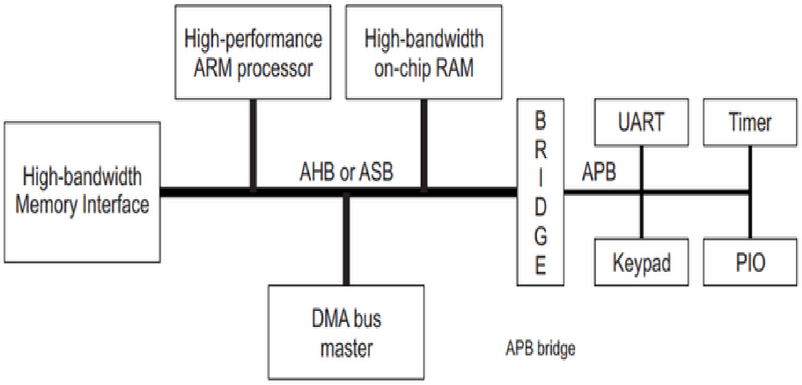
(d) Can we have only one slave agent in the abstract design? What will be the merits/demerits of this approach in formal verification of the complete/whole design?

3. Communication systems invariably deploy some kind of parity checking mechanisms for error detection and error correction. Consider an odd parity system with 4 input bits and one output bit. By definition, this system provides output 1 when an even number of inputs are 1. It is known to you that Reduced Ordered Binary Decision Diagrams (ROBDD) are a canonical way of formal representation of various designs for formal verification. Derive (step-by-step) ROBDD for this odd parity system. How this ROBDD changes in the case of an even parity system? [3 + 1]

4. Examine if the following FSM can serve as a model of a serial-adder? If yes, what is the meaning of the states s0 and s1 (i.e., what do they represent)? [2 + 1]



5. To connect multiple peripherals to a CPU, an interconnection protocol is utilized in the mechanism as shown below. Suppose, we wish to formally verify the Advanced Peripheral Bus (APB) protocol for any design bugs. You are supplied with a separate page that describes in detail the working of APB protocol. What would be your approach to carry out this verification? Explain your approach detailing all the steps involved? What are the drawbacks of your proposed solution? [3 + 4 + 2]


High-performance ARM processor
High-bandwidth on-chip RAM
High-bandwidth Memory Interface
AHB or ASB
DMA bus master
BRIDGE
APB
UART
Timer
Keypad
PIO
APB bridge

Essentially, your task is to only check the portion from bridge towards the right. In the above figure, you can notice that APB is helping connect UART (Universal Asynchronous Receiver Transmitter), Timer, Keyboard and Peripheral Input Output (PIO) with one another and the bridge. The bridge subsequently connects these components with the ARM processor and the random access memory (RAM).

6. What have you learnt from this course? Please do not mention the course contents, rather explain (in few statements) what skills/new knowledge you have gained from this course? [2]

APB PROTOCOL SPECIFICATIONS

APB is a simple bus mainly targeted for peripherals connections. Although the bus protocol does not have a bus width limitation, the common practice for ARM-based systems is to use a 32-bit peripheral bus. The latest version of APB is v2.0, which was a part of AMBA 4 realease. It is a low-cost interface and it is optimized for minimal power consumption and reduced interface complexity. Unlike AHB, it is a non-pipelined protocol, used to connect low-bandwidth peripherals. Mostly, used to connect the external peripheral to the SOC. In APB, every transfer takes at least two clock cycles (SETUP Cycle and ACCESS Cycle) to complete. It can also interface with AHB and AXI protocols using the bridges in between. Primarily, this protocol works as per the below FSM:

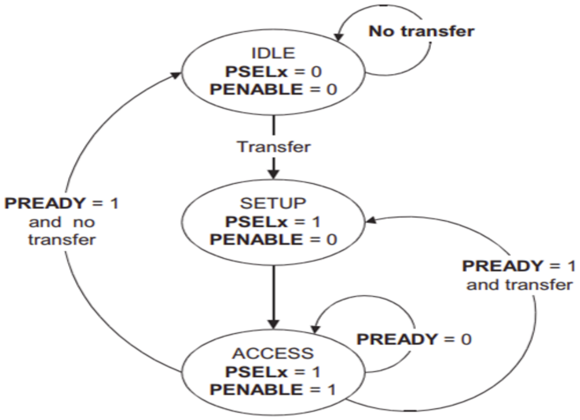
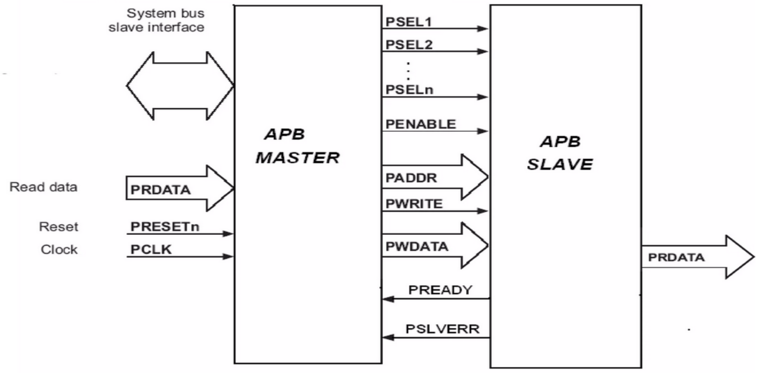


Figure 1: APB controller FSM Illustration

- IDLE: This is the default state of APB.
- SETUP: When transfer is required, PSELx is asserted then the bus moves in setup state. Bus only remains in SETUP for only one clock cycle and always moves to ACCESS state on next rising edge of clock. So, the slave must be able to sample the Address and control information in the SETUP cycle itself.
- ACCESS: PENABLE is asseted to enter into the ACCESS state. The PADDR, PWRITE, PSELx and PWDATA signals must remain stable during ACCESS state.

The organization of different signals between the master and slave can be shown as below:



Read operation under the APB protocol is shown as per the below timing waveform: Read operation is done as per the below events:

- At T1, a READ transfer with address PADDR, PWRITE and PSEL starts. They will be registered at rising edge of PCLK. This is SETUP Phase of the transfer.

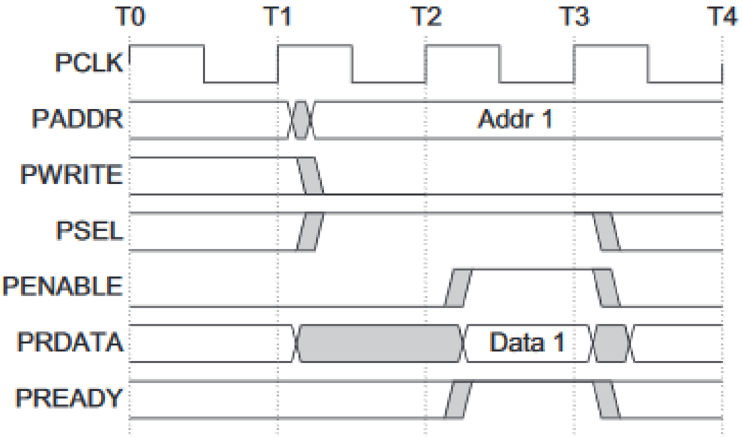


Figure 2: Read operation timing diagram

- After T2, PENABLE and PREADY are registered at the rising edge of PCLK.
- When asserted, PENABLE indicates the starting of ACCESS phase.
- When asserted, PREADY indicates that slave can complete the transfer at next rising edge of PCLK by providing the data on PRDATA.
- Slave must provide the data before the end of read transfer. i.e. before T3.

Write operation under the APB protocol is shown as per the below timing waveform:

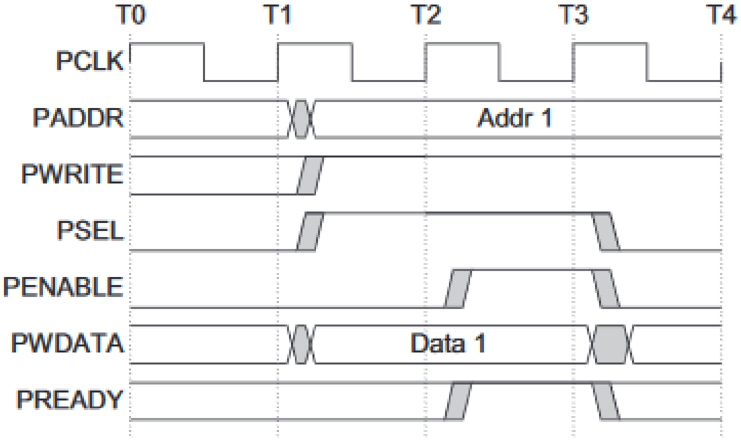


Figure 3: Write operation timing diagram

- At T1, a write transfer with address PADDR,PWDATA,PWRITE and PSEL starts. They will registered at the next rising edge of PCLK, T2. This is Setup Phase of Transfer.
- After T2, PENABLE and PREADY are registered at the rising edge of PCLK.
- When asserted, PENABLE indicates starting of ACCESS Phase
- When asserted, PREADY indicates that slave can complete the transfer at the next rising edige of PCLK.
- PADDR, PDATA and control signals all should remain valid till the transfer completes at T3. PENABLE signal will be de-asserted at the end of transfer.