UNIVERSITY OF TEHRAN
Electrical and Computer Engineering Department
Digital System Design with Hardware Description Languages
Fall 1394 / Test 1

Computer Account#_____

First Name:_____

Last Name:_____

Number:_____

Signature:_____

Grade:

Problem 1. _____/

Problem 2. _____/

Problem 3. _____/

Problem 4. _____/

Total: _____/ 100

THIS IS AN OPEN BOOK OPEN NOTE EXAM
YOU HAVE EXACTLY TWO HOURS AND A HALF TO DO THIS TEST
YOU MUST SHOW COMPLETE WORK ON ALL PROBLEMS

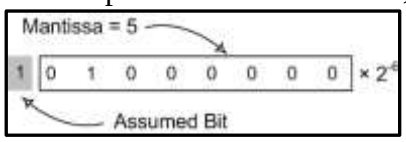
1. **RTL Implementation:** An arbiter has eight request lines and eight grant lines (req_i, and gnt_i). A requesting device holds its request active for as long as grant is not issued, and after grant is issued and the requesting device is using the arbitrated service or recourse. Priorities are set according to the most recent usage of the arbitrated service, where the device given the grant will be put at the end of the priority list (lowest in priority). Show datapath and controller for the arbiter, and VHDL description for the complete system.

2. **System Level:** A resolution function is to be written for assigning grants for several requesting devices. A requesting device indicates that it needs a certain service by announcing its non-zero integer id to the arbiter along with the time it needs to use the service (in US, Time). The arbiter issues grant to the device needing the service for the shortest time. The requesting device disconnects itself from the resolution function when it no longer needs the service. **A)** Write the arbitration resolution function. **B)** Write the VHDL code for the requesting device requesting service, **C)** Show VHDL code for the requesting device using the service after grant has been issued and usage of service for the time requested.

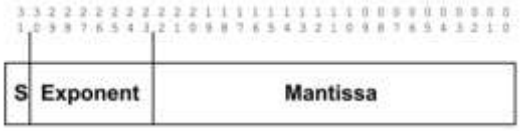
3. **RTL Design.** A floating point number in a given Radix has a Mantissa, an Exponent and a Sign. The value of the number depends on its Sign, Mantissa, Radix and Exponent, according to the following expression.

$$(Sign, Mantissa) \times Radix^{Exponent}$$

Mantissa is a normalized integer. Normalization is done such that the integer representing the Mantissa has no leading zeros resulting in more bits for representing the value of the number, and thus a better precision. Furthermore, since normalization removes all leading zeros, the left-most bit of a normalized number is always a 1. To save space, the left-most 1 of the Mantissa is assumed and never explicitly specified. With this arrangement, a normalized 8-bit Mantissa of value 5 is represented as shown here.



To normalize a number, its mantissa is shifted to the left enough places such that its left-most 1 is shifted out. To compensate for this shifting, the exponent is reduced by the number of shifts. The standard 32-bit IEEE format has a Sign bit, an 8-bit Exponent, and a 23-bit Mantissa as shown.



Single-Precision 32-bit format:
S: 0 or 1
Exponent: 8-bit excess-127
Mantissa: 23 bit normalized

Show datapath and controller for a 32-bit floating point adder. The operands are taken after a complete pulse on *start*. The inputs are normalized and the result must also be normalized. Use a barrel shifter. Disregard special cases.

4. Given the following signal assignments, show the exact transactions and events that expire on each signal. Show timing diagram for time 0 to 300 NS.

```
ENTITY HLsignals IS END HLsignals;
--
ARCHITECTURE dataflow OF HLsignals IS

TYPE HL IS ('L', '0', '1', 'H');
SIGNAL a, b, c : HL;

BEGIN

a <= '1' AFTER 20 NS,'0' AFTER 120 NS, 'H' AFTER 160 NS;
b <= HL'RIGHTOF(b) AFTER 25 NS
     WHEN (b /= HL'RIGHT AND NOW <= 70 NS)
     ELSE '0' AFTER 120 NS;
c <= '1', b AFTER 30 NS, a AFTER 50 NS,'H' AFTER 60 NS;

END dataflow;
```