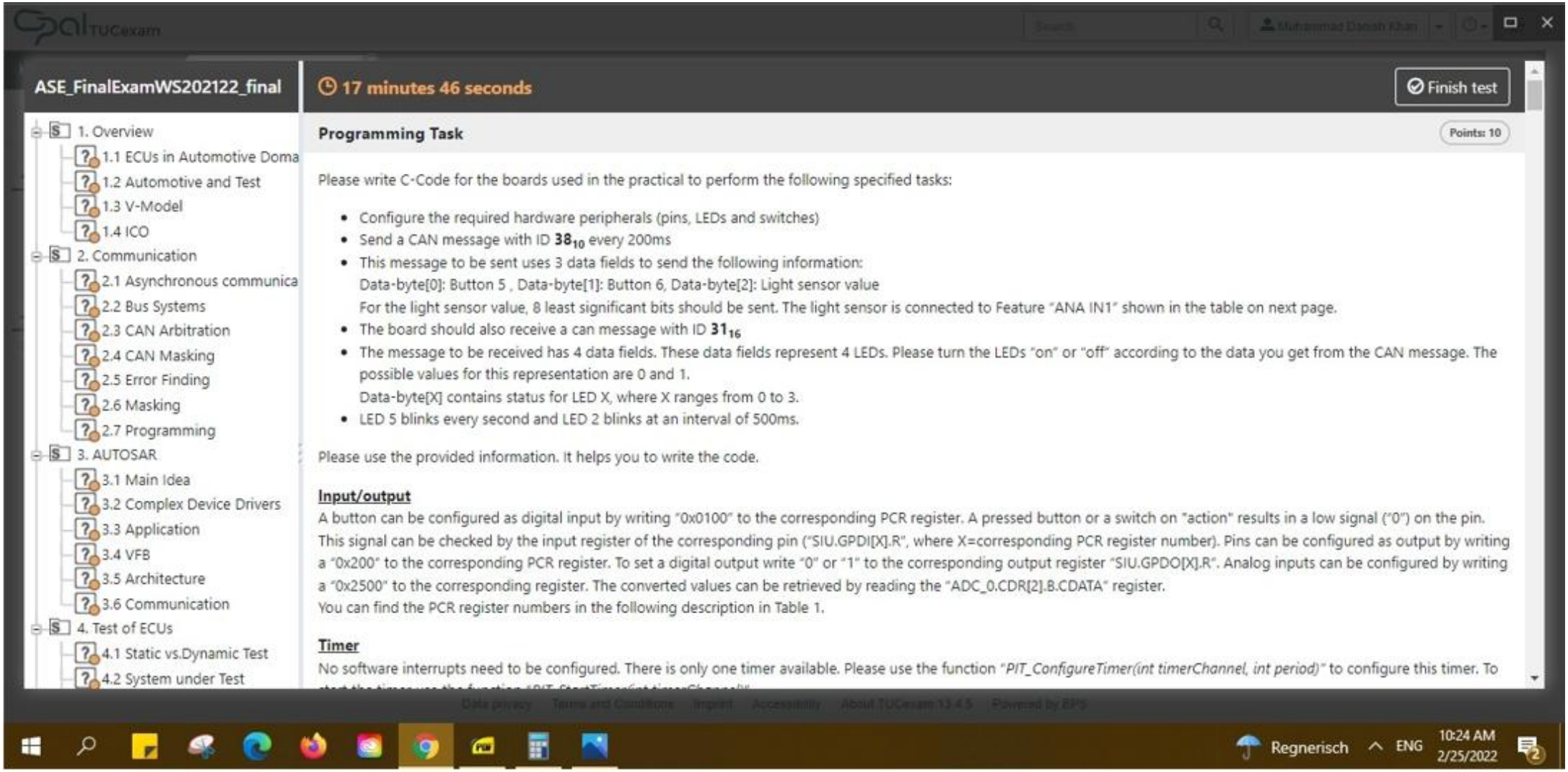
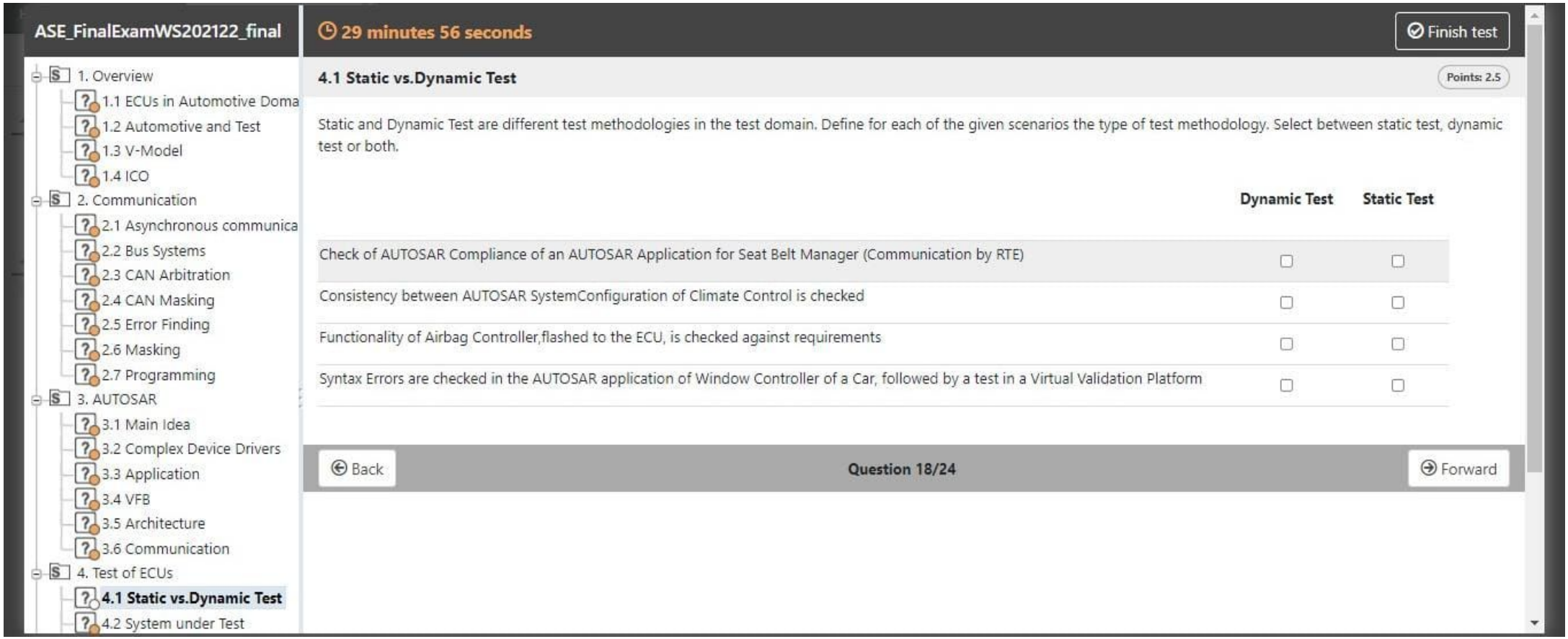

TUCexam
ASE_FinalExamWS202122_final
17 minutes 46 seconds
Finish test
1. Overview
1.1 ECUs in Automotive Doma
1.2 Automotive and Test
1.3 V-Model
1.4 ICO
2. Communication
2.1 Asynchronous communica
2.2 Bus Systems
2.3 CAN Arbitration
2.4 CAN Masking
2.5 Error Finding
2.6 Masking
2.7 Programming
3. AUTOSAR
3.1 Main Idea
3.2 Complex Device Drivers
3.3 Application
3.4 VFB
3.5 Architecture
3.6 Communication
4. Test of ECUs
4.1 Static vs.Dynamic Test
4.2 System under Test
Programming Task
Points: 10
Please write C-Code for the boards used in the practical to perform the following specified tasks:
Configure the required hardware peripherals (pins, LEDs and switches)
Send a CAN message with ID 38₁₀ every 200ms
This message to be sent uses 3 data fields to send the following information:
Data-byte[0]: Button 5 , Data-byte[1]: Button 6, Data-byte[2]: Light sensor value
For the light sensor value, 8 least significant bits should be sent. The light sensor is connected to Feature "ANA IN1" shown in the table on next page.
The board should also receive a can message with ID 31₁₆
The message to be received has 4 data fields. These data fields represent 4 LEDs. Please turn the LEDs "on" or "off" according to the data you get from the CAN message. The possible values for this representation are 0 and 1.
Data-byte[X] contains status for LED X, where X ranges from 0 to 3.
LED 5 blinks every second and LED 2 blinks at an interval of 500ms.
Please use the provided information. It helps you to write the code.
Input/output
A button can be configured as digital input by writing "0x0100" to the corresponding PCR register. A pressed button or a switch on "action" results in a low signal ("0") on the pin. This signal can be checked by the input register of the corresponding pin ("SIU.GPDI[X].R", where X=corresponding PCR register number). Pins can be configured as output by writing a "0x200" to the corresponding PCR register. To set a digital output write "0" or "1" to the corresponding output register "SIU.GPDO[X].R". Analog inputs can be configured by writing a "0x2500" to the corresponding register. The converted values can be retrieved by reading the "ADC_0.CDR[2].B.CDATA" register.
You can find the PCR register numbers in the following description in Table 1.
Timer
No software interrupts need to be configured. There is only one timer available. Please use the function "PIT_ConfigureTimer(int timerChannel, int period)" to configure this timer. To
Regnerisch
ENG
10:24 AM
2/25/2022


ASE_FinalExamWS202122_final
29 minutes 56 seconds
Finish test
1. Overview
1.1 ECUs in Automotive Doma
1.2 Automotive and Test
1.3 V-Model
1.4 ICO
2. Communication
2.1 Asynchronous communica
2.2 Bus Systems
2.3 CAN Arbitration
2.4 CAN Masking
2.5 Error Finding
2.6 Masking
2.7 Programming
3. AUTOSAR
3.1 Main Idea
3.2 Complex Device Drivers
3.3 Application
3.4 VFB
3.5 Architecture
3.6 Communication
4. Test of ECUs
4.1 Static vs.Dynamic Test
4.2 System under Test
4.1 Static vs.Dynamic Test
Points: 2.5
Static and Dynamic Test are different test methodologies in the test domain. Define for each of the given scenarios the type of test methodology. Select between static test, dynamic test or both.
Dynamic Test
Static Test
Check of AUTOSAR Compliance of an AUTOSAR Application for Seat Belt Manager (Communication by RTE)
Consistency between AUTOSAR SystemConfiguration of Climate Control is checked
Functionality of Airbag Controller,flashed to the ECU, is checked against requirements
Syntax Errors are checked in the AUTOSAR application of Window Controller of a Car, followed by a test in a Virtual Validation Platform
Back
Question 18/24
Forward

ASE_FinalExamWS202122_final

28 minutes 54 seconds

Finish test

- 1. Overview
  - 1.1 ECUs in Automotive Doma
  - 1.2 Automotive and Test
  - 1.3 V-Model
  - 1.4 ICO
- 2. Communication
  - 2.1 Asynchronous communica
  - 2.2 Bus Systems
  - 2.3 CAN Arbitration
  - 2.4 CAN Masking
  - 2.5 Error Finding
  - 2.6 Masking
  - 2.7 Programming
- 3. AUTOSAR
  - 3.1 Main Idea
  - 3.2 Complex Device Drivers
  - **3.3 Application**
  - 3.4 VFB
  - 3.5 Architecture
  - 3.6 Communication
- 4. Test of ECUs
  - 4.1 Static vs.Dynamic Test

3.3 Application

Points: 1

Which elements are used to create a software architecture (application) in AUTOSAR?

- ○ Webservice, FIBEX, SVN
- ○ Micro Services, Kubernetes, Cloud
- ○ AI, Machine Learning, SVM
- ○ Function, Classes, , Registers, Interrupts
- ◉ Software components, Ports, Interfaces, Runnables
- ○ Container, Constructor, Loops
- ○ Software components, Database, Registers

Back

Question 14/24

Forward

ASE_FinalExamWS202122_final

28 minutes 36 seconds

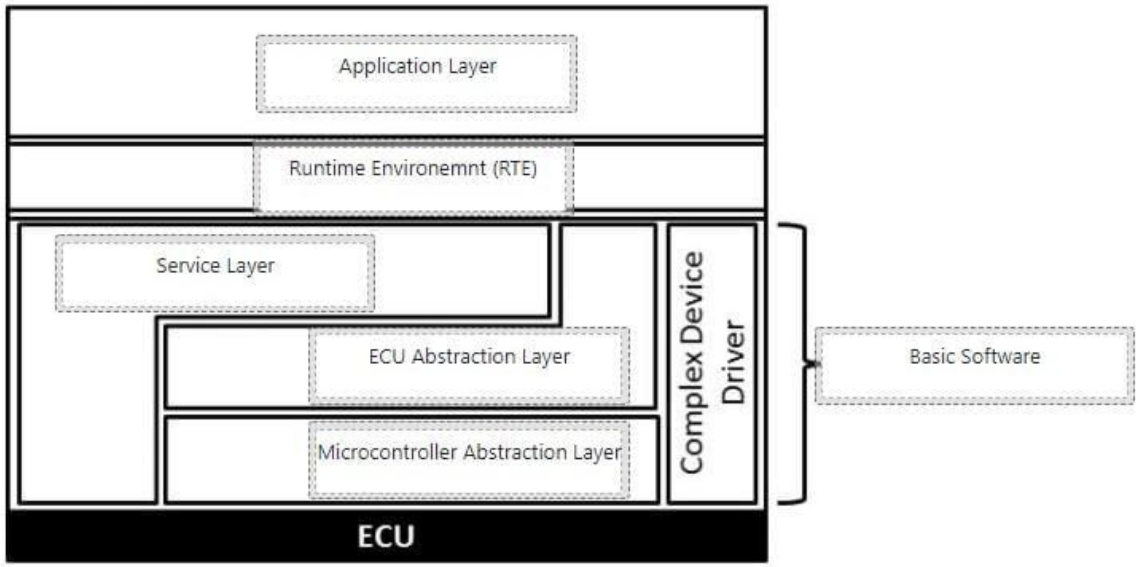
Finish test

- 1. Overview
  - 1.1 ECUs in Automotive Doma
  - 1.2 Automotive and Test
  - 1.3 V-Model
  - 1.4 ICO
- 2. Communication
  - 2.1 Asynchronous communica
  - 2.2 Bus Systems
  - 2.3 CAN Arbitration
  - 2.4 CAN Masking
  - 2.5 Error Finding
  - 2.6 Masking
  - 2.7 Programming
- 3. AUTOSAR
  - 3.1 Main Idea
  - 3.2 Complex Device Drivers
  - 3.3 Application
  - 3.4 VFB
  - **3.5 Architecture**
  - 3.6 Communication
- 4. Test of ECUs
  - 4.1 Static vs.Dynamic Test

3.5 Architecture

Points: 3

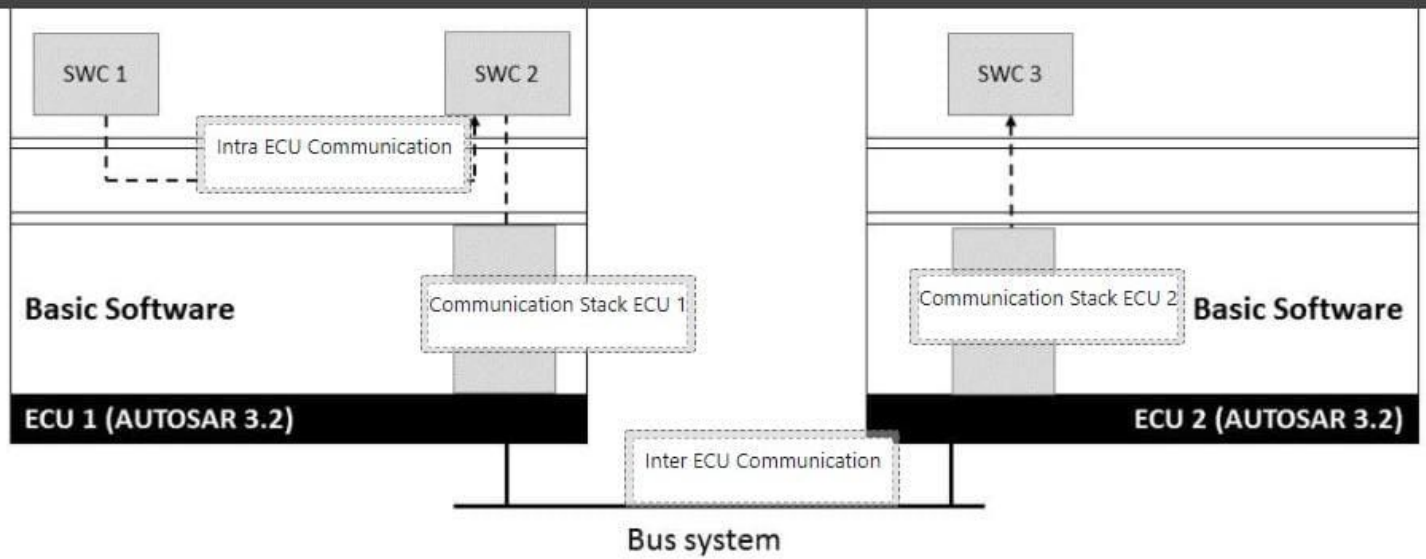
Put the words to the correct position in the architecture picture of AUTOSAR.



- 1.2 Automotive and Test
- 1.3 V-Model
- 1.4 ICO
- 2. Communication
  - 2.1 Asynchronous communica
  - 2.2 Bus Systems
  - 2.3 CAN Arbitration
  - 2.4 CAN Masking
  - 2.5 Error Finding
  - 2.6 Masking
  - 2.7 Programming
- 3. AUTOSAR
  - 3.1 Main Idea
  - 3.2 Complex Device Drivers
  - 3.3 Application
  - 3.4 VFB
  - 3.5 Architecture
  - **3.6 Communication**
- 4. Test of ECUs
  - 4.1 Static vs.Dynamic Test
  - 4.2 System under Test
  - 4.3 Test and V-Model
  - 4.4 Simulation Types
  - 4.5 Test Parking Brake

28 minutes 24 seconds

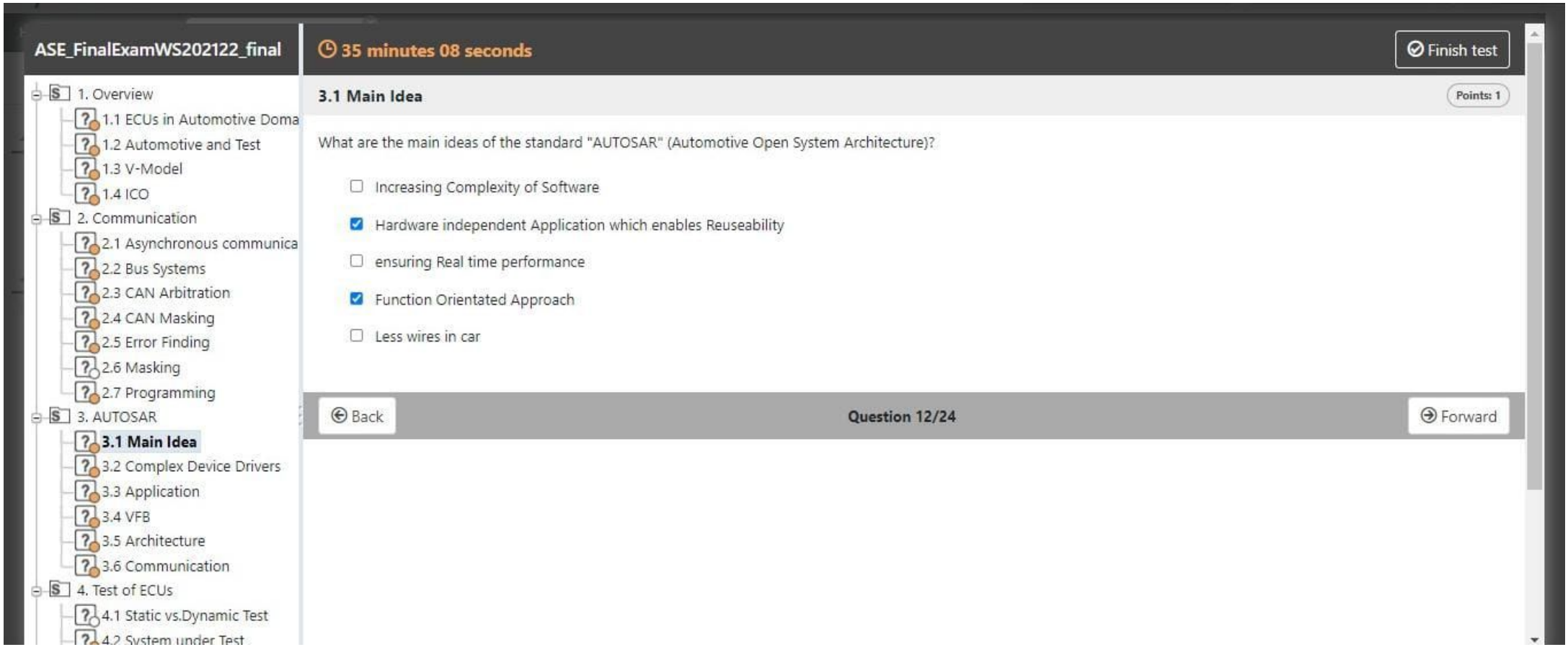
Finish test

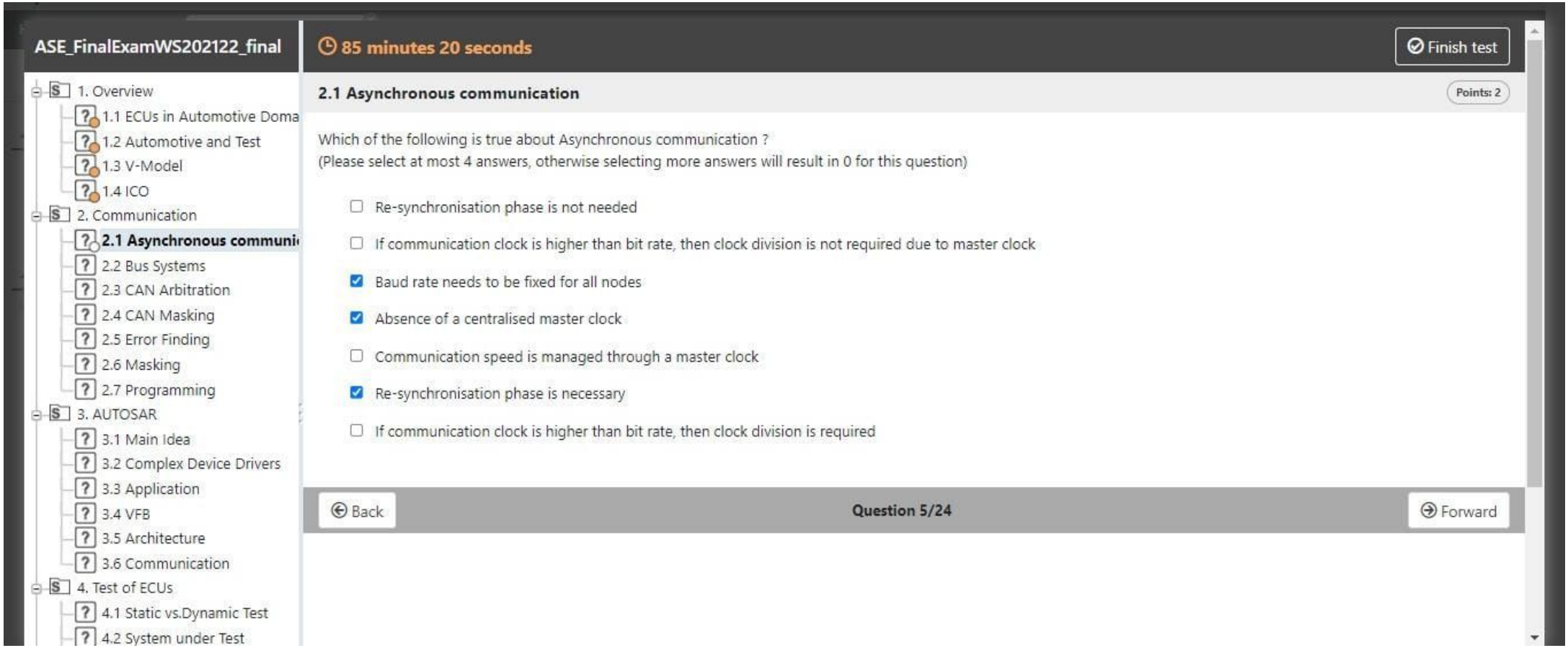


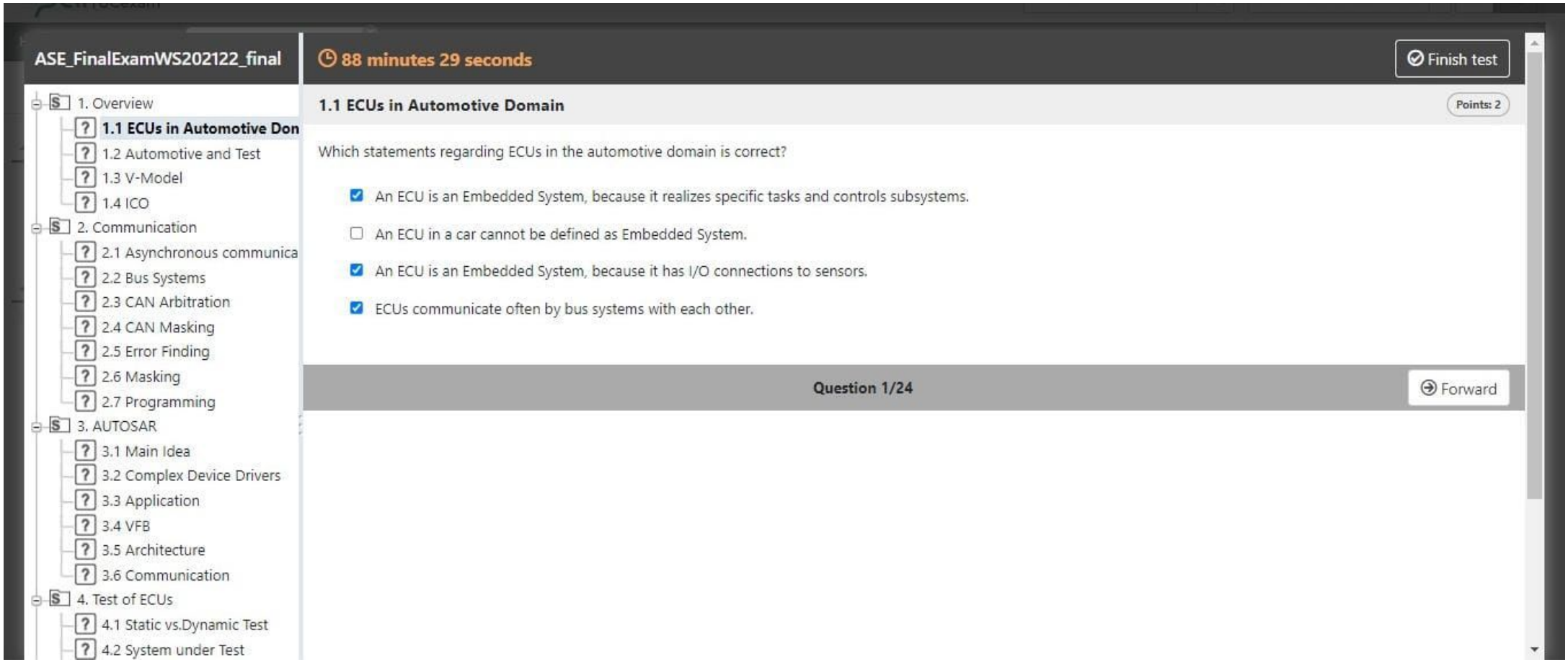
Back

Question 17/24

Forward


ASE_FinalExamWS202122_final
35 minutes 08 seconds
Finish test
1. Overview
1.1 ECUs in Automotive Doma
1.2 Automotive and Test
1.3 V-Model
1.4 ICO
2. Communication
2.1 Asynchronous communica
2.2 Bus Systems
2.3 CAN Arbitration
2.4 CAN Masking
2.5 Error Finding
2.6 Masking
2.7 Programming
3. AUTOSAR
3.1 Main Idea
3.2 Complex Device Drivers
3.3 Application
3.4 VFB
3.5 Architecture
3.6 Communication
4. Test of ECUs
4.1 Static vs.Dynamic Test
4.2 System under Test
3.1 Main Idea
Points: 1
What are the main ideas of the standard "AUTOSAR" (Automotive Open System Architecture)?
Increasing Complexity of Software
Hardware independent Application which enables Reuseability
ensuring Real time performance
Function Orientated Approach
Less wires in car
Back
Question 12/24
Forward


ASE_FinalExamWS202122_final
85 minutes 20 seconds
Finish test
1. Overview
1.1 ECUs in Automotive Doma
1.2 Automotive and Test
1.3 V-Model
1.4 ICO
2. Communication
2.1 Asynchronous communi
2.2 Bus Systems
2.3 CAN Arbitration
2.4 CAN Masking
2.5 Error Finding
2.6 Masking
2.7 Programming
3. AUTOSAR
3.1 Main Idea
3.2 Complex Device Drivers
3.3 Application
3.4 VFB
3.5 Architecture
3.6 Communication
4. Test of ECUs
4.1 Static vs.Dynamic Test
4.2 System under Test
2.1 Asynchronous communication
Points: 2
Which of the following is true about Asynchronous communication ?
(Please select at most 4 answers, otherwise selecting more answers will result in 0 for this question)
Re-synchronisation phase is not needed
If communication clock is higher than bit rate, then clock division is not required due to master clock
Baud rate needs to be fixed for all nodes
Absence of a centralised master clock
Communication speed is managed through a master clock
Re-synchronisation phase is necessary
If communication clock is higher than bit rate, then clock division is required
Back
Question 5/24
Forward


ASE_FinalExamWS202122_final
88 minutes 29 seconds
Finish test
1. Overview
1.1 ECUs in Automotive Don
1.2 Automotive and Test
1.3 V-Model
1.4 ICO
2. Communication
2.1 Asynchronous communica
2.2 Bus Systems
2.3 CAN Arbitration
2.4 CAN Masking
2.5 Error Finding
2.6 Masking
2.7 Programming
3. AUTOSAR
3.1 Main Idea
3.2 Complex Device Drivers
3.3 Application
3.4 VFB
3.5 Architecture
3.6 Communication
4. Test of ECUs
4.1 Static vs.Dynamic Test
4.2 System under Test
1.1 ECUs in Automotive Domain
Points: 2
Which statements regarding ECUs in the automotive domain is correct?
An ECU is an Embedded System, because it realizes specific tasks and controls subsystems.
An ECU in a car cannot be defined as Embedded System.
An ECU is an Embedded System, because it has I/O connections to sensors.
ECUs communicate often by bus systems with each other.
Question 1/24
Forward

ASE_FinalExamWS202122_final

87 minutes 41 seconds

Finish test

- 1. Overview
  - 1.1 ECUs in Automotive Doma
  - **1.2 Automotive and Test**
  - 1.3 V-Model
  - 1.4 ICO
- 2. Communication
  - 2.1 Asynchronous communica
  - 2.2 Bus Systems
  - 2.3 CAN Arbitration
  - 2.4 CAN Masking
  - 2.5 Error Finding
  - 2.6 Masking
  - 2.7 Programming
- 3. AUTOSAR
  - 3.1 Main Idea
  - 3.2 Complex Device Drivers
  - 3.3 Application
  - 3.4 VFB
  - 3.5 Architecture
  - 3.6 Communication
- 4. Test of ECUs
  - 4.1 Static vs.Dynamic Test
  - 4.2 System under Test

1.2 Automotive and Test

Points: 2

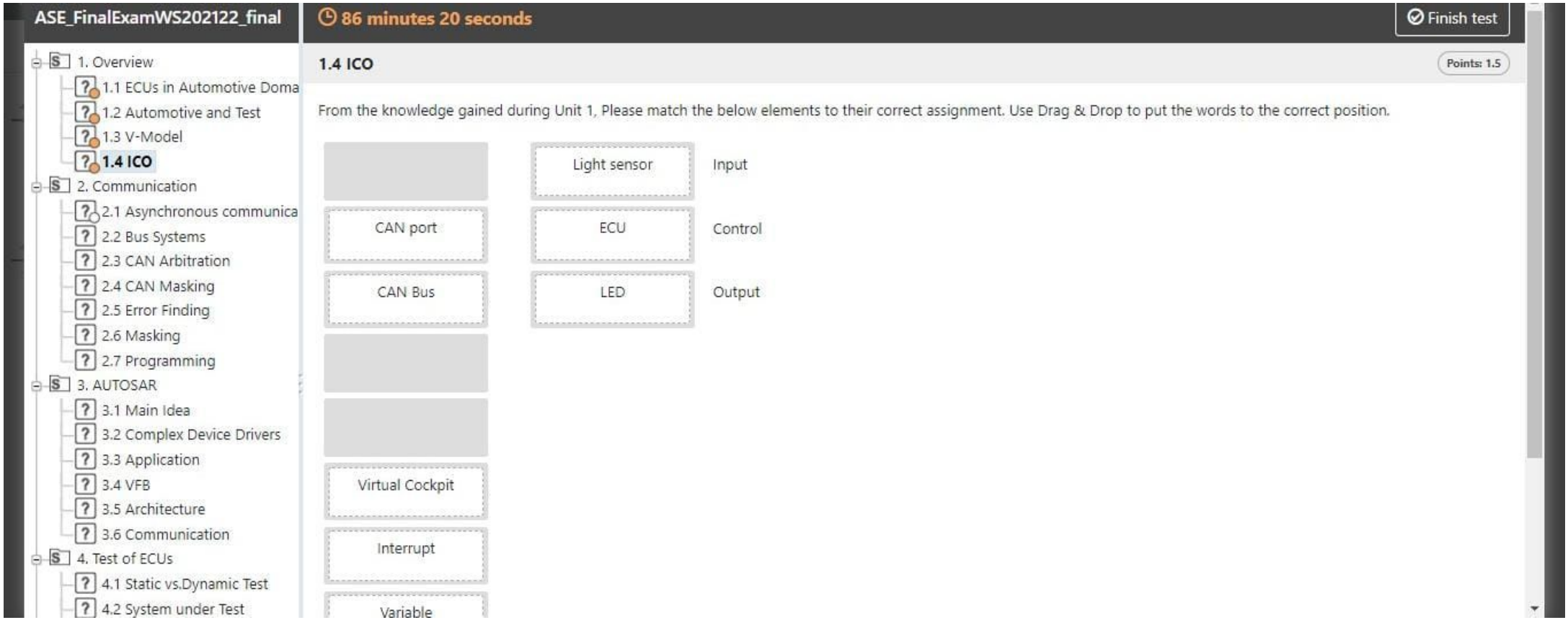
Is it necessary to test ECUs in the automotive domain? Explain your answer briefly.

ECUs should be tested mandatorily. ECU integrates functionalities for calculation, analysis of sensor data and controlling actuators.

White box testing should be performed in calculation and on sensed data provided by the dedicated sensor for the ECU. Black box test should be performed in the

Current word count: 50

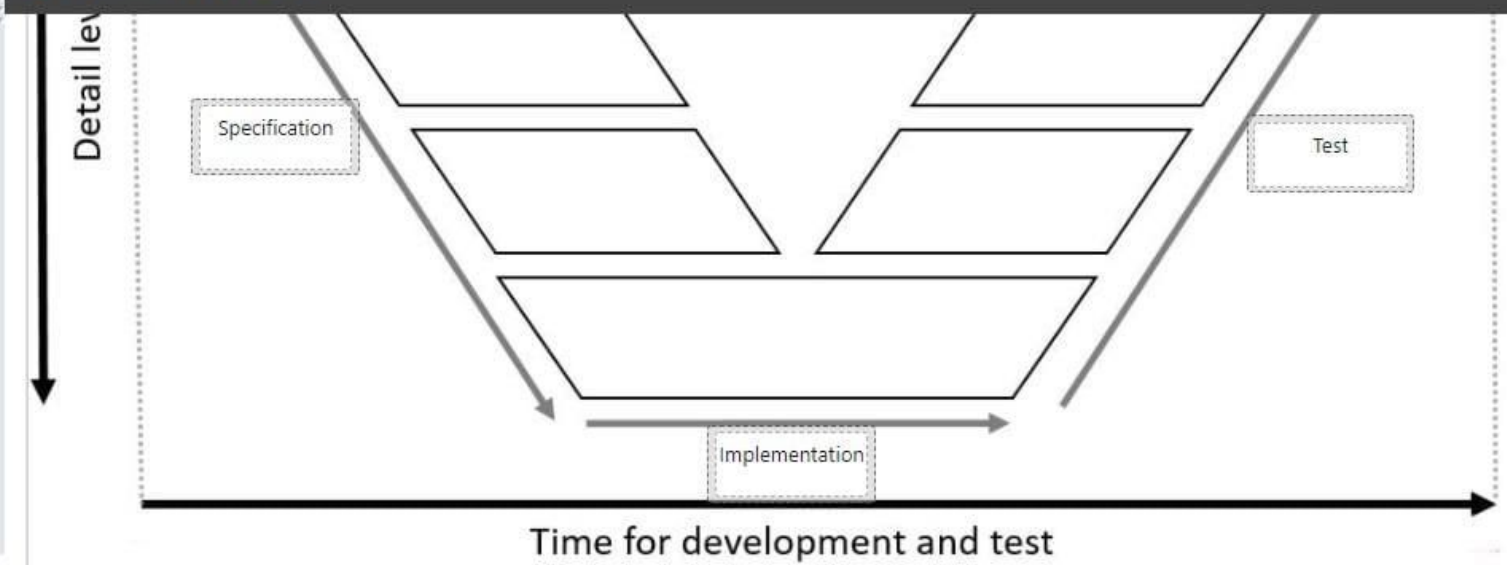
Back | **Question 2/24** | Forward


ASE_FinalExamWS202122_final
86 minutes 20 seconds
Finish test
1. Overview
1.1 ECUs in Automotive Doma
1.2 Automotive and Test
1.3 V-Model
1.4 ICO
2. Communication
2.1 Asynchronous communica
2.2 Bus Systems
2.3 CAN Arbitration
2.4 CAN Masking
2.5 Error Finding
2.6 Masking
2.7 Programming
3. AUTOSAR
3.1 Main Idea
3.2 Complex Device Drivers
3.3 Application
3.4 VFB
3.5 Architecture
3.6 Communication
4. Test of ECUs
4.1 Static vs.Dynamic Test
4.2 System under Test
1.4 ICO
Points: 1.5
From the knowledge gained during Unit 1, Please match the below elements to their correct assignment. Use Drag & Drop to put the words to the correct position.
Light sensor
Input
CAN port
ECU
Control
CAN Bus
LED
Output
Virtual Cockpit
Interrupt
Variable

- 2.7 Programming
- 3. AUTOSAR
  - 3.1 Main Idea
  - 3.2 Complex Device Drivers
  - 3.3 Application
  - 3.4 VFB
  - 3.5 Architecture
  - 3.6 Communication
- 4. Test of ECUs
  - 4.1 Static vs.Dynamic Test
  - 4.2 System under Test
  - 4.3 Test and V-Model
  - 4.4 Simulation Types
  - 4.5 Test Parking Brake
  - 4.6 Test Seat Heating
- 5. Programming
  - Programming Task

87 minutes 08 seconds

Finish test



ASE_FinalExamWS202122_final

9 minutes 26 seconds

Finish test

- 1. Overview
  - 1.1 ECUs in Automotive Doma
  - 1.2 Automotive and Test
  - 1.3 V-Model
  - 1.4 ICO
- 2. Communication
  - 2.1 Asynchronous communica
  - 2.2 Bus Systems
  - 2.3 CAN Arbitration
  - 2.4 CAN Masking
  - 2.5 Error Finding
  - **2.6 Masking**
  - 2.7 Programming
- 3. AUTOSAR
  - 3.1 Main Idea
  - 3.2 Complex Device Drivers
  - 3.3 Application
  - 3.4 VFB
  - 3.5 Architecture
  - 3.6 Communication
- 4. Test of ECUs
  - 4.1 Static vs.Dynamic Test

2.6 Masking

Points: 2.5

Following is an 4-bit MASK and ACCEPT configuration. Please select which of the IDs can be accepted or not, using this configuration on an ECU node connected to a CAN-Bus.

Mask : 1000
ACCEPT: 1XXX

	Will be Rejected	Will be Accepted
0110	◉	○
XXXX (any)	◉	○
1101	○	◉
0x2	◉	○
0xF	○	◉

Back

Question 10/24

Forward

ASE_FinalExamWS202122_final

68 minutes 28 seconds

Finish test

- 1. Overview
  - 1.1 ECUs in Automotive Doma
  - 1.2 Automotive and Test
  - 1.3 V-Model
  - 1.4 ICO
- 2. Communication
  - 2.1 Asynchronous communica
  - 2.2 Bus Systems
  - 2.3 CAN Arbitration
  - 2.4 CAN Masking
  - **2.5 Error Finding**
  - 2.6 Masking
  - 2.7 Programming
- 3. AUTOSAR
  - 3.1 Main Idea
  - 3.2 Complex Device Drivers
  - 3.3 Application
  - 3.4 VFB
  - 3.5 Architecture
  - 3.6 Communication
- 4. Test of ECUs
  - 4.1 Static vs.Dynamic Test
  - 4.2 System under Test

2.5 Error Finding

Points: 4

Following is a function for sending a 16 bit long unsigned counter value encapsulated in a can message with id 0x10. The function is called at a time interval of every 500 ms and should send an incremented value of "counter" variable by 30 at every time interval. Upon reaching a value that equals 1000 or more, the counter should be reset to zero. The variable must be split in two parts of 8 bits each, in order to accommodate it in a can message. The first data byte of the can message should contain the first 8 least significant bits of the aforementioned variable "counter". And the second data byte should contain 8 most significant bits. Analyse the following piece of code to mark and select the correction option for the problems/errors against the above specified requirements in the below function:

```
void counterUpdateSend()
{
static uint16 counter = 0; // variable declaration for counter   A No error
counter = counter + 30; // counter increment
if (counter = 1000)   B Incorrect condition
{ counter = 0; }
CAN_0.BUF[8].MSG_ID.B.STD_ID = 10; // CAN message id   B Error
CAN_0.BUF[8].CS.B.LENGTH = 1; // CAN message data length   B Error
CAN_0.BUF[8].DATA.B[0] = counter % 128; // obtaining first 8 least significant bits from the variable   A No error
CAN_0.BUF[8].DATA.B[0] = counter / 256; // obtaining the 8 most significant bits   A No error
CAN_0.BUF[8].CS.B.CODE = 0xC; // buffer code for transmitting a message   A No error
}
```

ASE_FinalExamWS202122_final

62 minutes 57 seconds

Finish test

- 1. Overview
  - 1.1 ECUs in Automotive Doma
  - 1.2 Automotive and Test
  - 1.3 V-Model
  - 1.4 ICO
- 2. Communication
  - 2.1 Asynchronous communica
  - 2.2 Bus Systems
  - 2.3 CAN Arbitration
  - 2.4 CAN Masking
  - 2.5 Error Finding
  - 2.6 Masking
  - 2.7 Programming
- 3. AUTOSAR
  - 3.1 Main Idea
  - 3.2 Complex Device Drivers
  - 3.3 Application
  - **3.4 VFB**
  - 3.5 Architecture
  - 3.6 Communication
- 4. Test of ECUs
  - 4.1 Static vs.Dynamic Test

3.4 VFB

Points: 2

What does VFB mean in the context of AUTOSAR? State its benefit and the software development stage in which it used.

Virtual Functional Bus (VFB) is responsible of realizing the communication between the software components, and validates the interaction of all components before actual software implementation. The VFB is realized by the RTE in the software.

Current word count: 35

Back | **Question 15/24** | Forward

ASE_FinalExamWS202122_final

60 minutes 57 seconds

Finish test

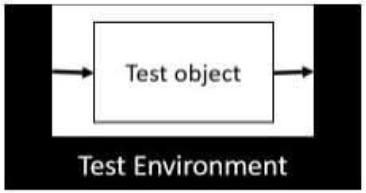
- 1. Overview
  - 1.1 ECUs in Automotive Doma
  - 1.2 Automotive and Test
  - 1.3 V-Model
  - 1.4 ICO
- 2. Communication
  - 2.1 Asynchronous communica
  - 2.2 Bus Systems
  - 2.3 CAN Arbitration
  - 2.4 CAN Masking
  - 2.5 Error Finding
  - 2.6 Masking
  - 2.7 Programming
- 3. AUTOSAR
  - 3.1 Main Idea
  - 3.2 Complex Device Drivers
  - 3.3 Application
  - 3.4 VFB
  - 3.5 Architecture
  - 3.6 Communication
- 4. Test of ECUs
  - 4.1 Static vs.Dynamic Test
  - 4.2 System under Test

4.2 System under Test

Points: 2

The SuT (System under Test) represents the test object and test environment.

What does **PoC** and **PoO** in SuT stand for? What do they realize?



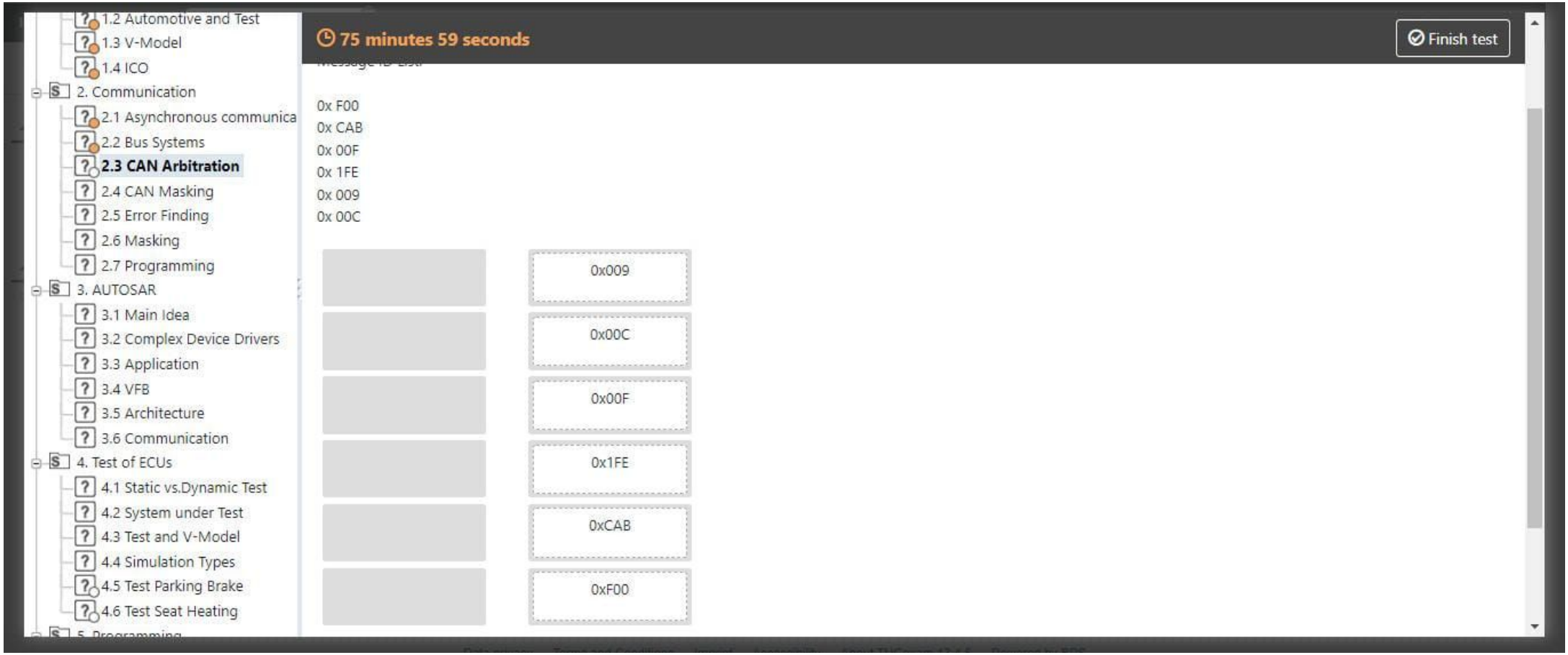
Point of Control (PoC) defines the point where Test object is stimulated with test data.
Point of Observation (PoO) defines the point where output of Test Object is read.

Current word count: 29

Back

Question 19/24

Forward


1.2 Automotive and Test
1.3 V-Model
1.4 ICO
2. Communication
2.1 Asynchronous communica
2.2 Bus Systems
2.3 CAN Arbitration
2.4 CAN Masking
2.5 Error Finding
2.6 Masking
2.7 Programming
3. AUTOSAR
3.1 Main Idea
3.2 Complex Device Drivers
3.3 Application
3.4 VFB
3.5 Architecture
3.6 Communication
4. Test of ECUs
4.1 Static vs.Dynamic Test
4.2 System under Test
4.3 Test and V-Model
4.4 Simulation Types
4.5 Test Parking Brake
4.6 Test Seat Heating
5. Programming
75 minutes 59 seconds
Finish test
0x F00
0x CAB
0x 00F
0x 1FE
0x 009
0x 00C
0x009
0x00C
0x00F
0x1FE
0xCAB
0xF00

TUCexam

Search

Abdul Sajid

ASE_FinalExamWS202122_final

45 minutes 31 seconds

Finish test

- 1. Overview
  - 1.1 ECUs in Automotive Doma
  - 1.2 Automotive and Test
  - 1.3 V-Model
  - 1.4 ICO
- 2. Communication
  - 2.1 Asynchronous communica
  - 2.2 Bus Systems
  - 2.3 CAN Arbitration
  - **2.4 CAN Masking**
  - 2.5 Error Finding
  - 2.6 Masking
  - 2.7 Programming
- 3. AUTOSAR
  - 3.1 Main Idea
  - 3.2 Complex Device Drivers
  - 3.3 Application
  - 3.4 VFB
  - 3.5 Architecture
  - 3.6 Communication
- 4. Test of ECUs
  - 4.1 Static vs.Dynamic Test
  - 4.2 System under Test

2.4 CAN Masking

Points: 2

The following is a snapshop of a message register that respresents an 8-digit message ID of a CAN-Message

ID bit 7	ID bit 6	ID bit 5	ID bit 4	ID bit 3	ID bit 2	ID bit 1	ID bit 0
MSB							LSB

Calculate the masking (MASK) and acceptance register (ACCEPT) values to approve only CAN messages with ID 229_{10} ($E05_{16}$) and 197_{10} ($C05_{16}$). (Numbers are given in decimal and numbers in brackets are given in Hexadecimal format)
(Please answer the question in Hexadecimal/Binary formats without any spaces only)

Mask:110111111111

Accept:00X000000101

Back

Question 8/24

Forward