Version 1.3 - 16/11/2022

Examination guide for FYS4220 2022

Document revision

Version	Date	Comment
1.1	29/11/2021	First version for 2021
1.2	10/12/2021	Updated question on SPI protocol for category B. Added the option to choose between UART or SPI.
1.3	16/11/2022	Updated for 2022 – no major changes, only small adjustements.

General information

To help you prepare for the oral exam we have made this examination guide that lists the five main categories with relevant questions and topics that will be covered during the exam:

A. Main project
B. General technology, programmable logic and FPGAs
C. VHDL
D. Embedded systems
E. Real-time operating systems

We would like to give you the opportunity to get a good start on the oral exam. We will therefore start the exam by discussing the embedded systems project. You will all be asked to cover the topics listed under category A, which you can find at the end of this document. We plan to spend about 10 minutes on this part.

For the other categories (B–E) a set of questions/topics are listed at the end of this document, and these will provide the basis for our discussion. We will not have time to cover every question, but all questions are subject to the discussion. We may also ask questions that are not listed in the examination guide, but this will only be done if we think this may help to improve your result. Where possible, we will use the main project as the starting point for the questions and topics in these categories.

The listed questions and topics cover the material and knowledge you are expected to have acquired from completing FYS4220.

In general we expect you to start talking about a topic/question in a way that demonstrates your knowledge and understanding of this topic/question. If we are happy about the way you present the material we will let you continue to talk as we guide you through the topics

we would like to discuss. This will give you an opportunity to influence the content and direction of the discussions.
When discussing code, we will focus on the understanding of concepts and overall purpose of the code. Accurate reproduction of detailed syntax will be given less importance. Although not giving full score, pseudo-code may be acceptable if the main understanding of the concept and purpose is convincingly demonstrated.

To help the discussion we may bring up visual aids such as figures, text or code that we may ask you to discuss or explain. This will be figures or code that are related to the questions in the examination guide, and that have been covered in the course material and therefore already should be familiar to you.

You should be prepared to draw or write during the exam. If the exam is performed in person, either the blackboard or a pen and paper will be used.

If the exam is performed online through Zoom you should make sure you also have an option to share your drawing or writing. If you have a tablet or PC with a touch screen or digital pen option, you can for example log in to the same Zoom meeting and share your screen. Alternatively you can also use your phone as a document camera, logging in to the same Zoom meeting, turning on the camera and mounting the found in a way that allows you to show a piece of paper that you are writing on. In the case we may ask you to write a short piece of code, you may use an editor on your PC that has syntax highlighting for c or VHDL. You can then either share you screen or editor window. But if you prefer, you can also use handwriting for this purpose. We recommend you to try these this solutions already now to be well prepared on the exam day.

More information on digital oral exams at UiO can be found here:
https://www.uio.no/studier/eksamen/digital-muntlig-og-mastereksamen/gjennomforing.html

The language of the exam will either be Norwegian or English depending on your choice. We know that the topics covered in this course make use of many words and expressions for which we may not have good or obvious alternatives in Norwegian. We will therefore of course accept that you use the English words and expressions even if you preferred language for the exam is Norwegian.

We tentatively plan to spend up to 45 minutes per candidate.

Version 1.3 - 16/11/2022

Discussion topics and questions

A. Project

We will start the examination by asking you to briefly present the embedded systems project that you have been working with this semester. The external examiner is not very familiar with the project and your task will be to take a top-down approach to:

- Briefly explain the main purpose of the system
- Explain how the project was split into different parts and explain the main objective and content of these parts.
- Briefly discuss the different aspects the project required you to work on (FPGA design, software, system, tools etc.).

A set of figures are provided at the end of this document that may be helpful during the discussion. You may also ask us to point to other figures in the project description on the course Github page.

B. General technology, programmable logic and FPGAs

- Explain the behaviour of the basic logic gates.
- Be able to explain the basic concept of programmable logic / FPGAs.
- Describe and explain the basic building block of an FPGA – the logic block.
- Explain the concept and purpose of the Look-up table (LUT).
- Be able to explain what the configuration memory of an FPGA is.
- Show a basic understanding of why synchronization of signals between clock domains are required (metastability, setup and hold times).
- Be able to explain the SPI or UART serial communication and message protocol.
- Name and briefly explain a few advantages and disadvantages that are related to the use of FPGAs

C. VHDL

Background and general concepts:

- Briefly explain what a hardware description language is and be able to say a few words about the history of VHDL.
- What is the main conceptual difference between writing an HDL description and writing software in a high-level language like C or python?
- Name and explain the purpose of the main design units of VHDL.
- Name and explain the description styles that can be used when writing a VHDL design.
- Briefly explain what is meant by RTL.
- What is meant by a data type and object in VHDL?
- Can you name a few commonly used data types?
- What does it mean that the VHDL language is strongly typed and why can this be an advantage?

- What is meant by a concurrent statement in VHDL?
- Explain the concept of concurrent and sequential statements in VHDL, and in which part of a VHDL description these can be use.
- Explain how edge detection on a logic signal can be implemented in VHDL.
- Briefly explain what a subprogram is in VHDL and for what purpose these can be used in the VHDL language.
- What is the purpose of the functions rising_edge() and falling_edge(), and how is their use limited?

Process:

- For what purposes can a process be used in VHDL?
- Explain the use of the sensitivity list for the various types of process descriptions.
- Be able to explain the difference between asynchronous and synchronous resets for a VHDL process and show how these are implemented in VHDL.

- What is meant by an incomplete sensitivity list in VHDL?
- What is meant by an unintentional latch / incomplete signal assignment in VHDL and how can it be avoided?
- Be able to write and explain the VHDL description for a basic process (e.g. d-flipflop with synchronous or asynchronous reset, with or without enable; a basic counter; or a multiplexer).
-

State machines:

- Briefly explain what a state machine is and for what purpose it is typically used.
- Explain how a state machine can be implemented in VHDL
    - Conceptually on a block diagram level (e.g. what type of logic is needed)
    - The VHDL statements that are needed to declare and describe the states of a state machine
    - The main process and conditional statements used to describe the functionality of a state machine.
- Explain the difference between a Moore and Mealy state machine

Test benches:

- For what purpose is a testbench used in VHDL?
- Explain the basic structure of a VHDL test bench.
- Explain how a process is used differently when writing the test bench stimuli process compared to a process for describing logic that will be synthesized and downloaded to an FPGA.
- Mention and briefly explain the different complexity levels that can be used when writing a VHDL test bench.
- Write a VHDL statement that can be used model a clock for the use in a test bench.
- Briefly explain what the UVVM utility library is and how this was used in the project assignment

D. Embedded systems

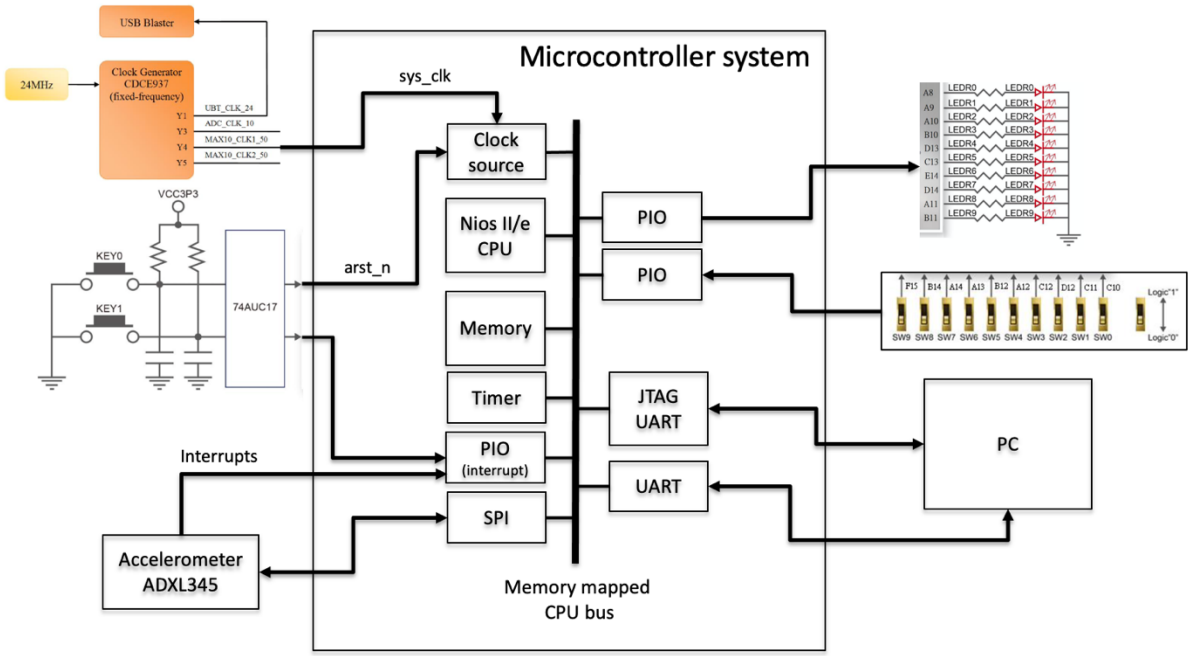
- Explain what is meant by an embedded system and how it is different from a general-purpose computer (PC).
- Name and briefly describe some of the main constraints and characteristics of an embedded system.
- Give examples of typical embedded system application areas-
- Briefly explain what a microcontroller is.
- Briefly explain what is meant by a System on Chip (SoC)
- What is meant by softcore and hardcore processor?
- What is the main difference between a Harvard type and Von Neumann type architecture?
- Explain the concept of memory mapped I/O.
- What is a memory mapped register interface and for what purpose was this used in the project assignment?
- Explain the basic concept of the hardware abstraction layer.
- What is a board support package?
- What are the macro functions IOWR and IORD used for?
- What is the purpose of a driver for an embedded system?

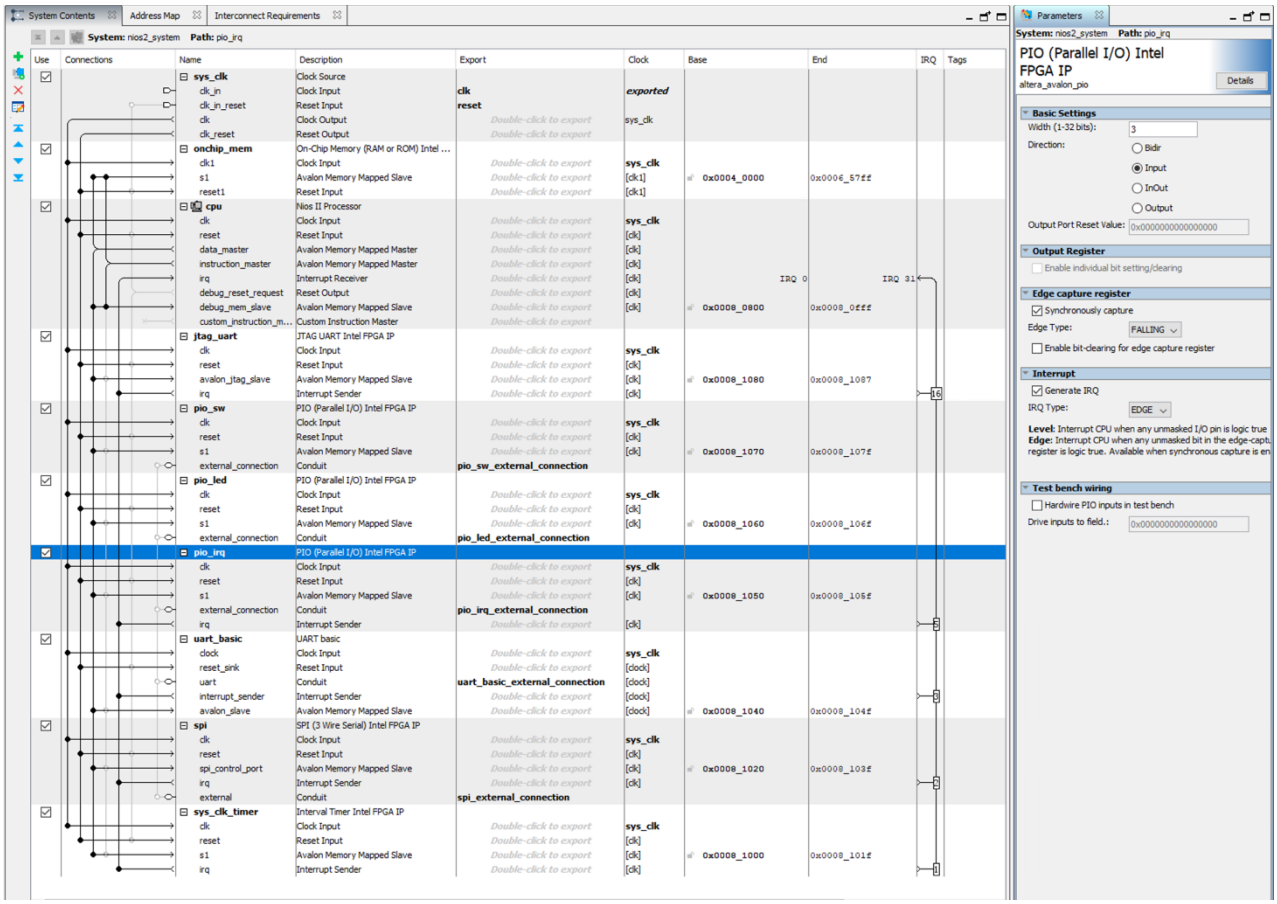
E. Real-time operating systems

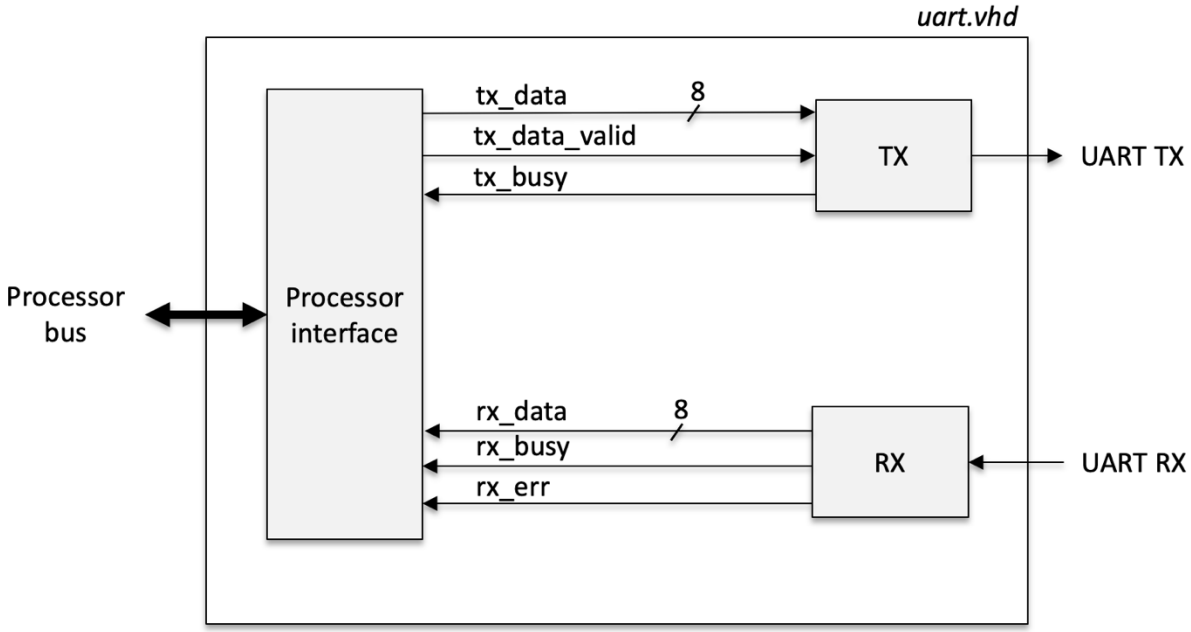
- Explain the main characteristic of a Real-time embedded system (i.e. compared to a standard embedded system)
- Explain how real-time systems can be divided into hard real-time systems and soft real-time systems.
- What is meant by time management and task management when talking about real-time systems?
- What is the purpose of the scheduler in a real-time system?
- What is meant by a task in a real-time system?
- Explain the main states of a task in a real-time system?
- What is meant by blocking and why is it necessary in a real-time system?
- Explain the concept of context switch.
- Explain what is meant by inter-task communication.
- Explain the main methods available in a real-time system to perform inter-task communication.
- Explain the concept of a semaphore and how and for what purpose it can be used in a real-time system.
- Explain the concept of message mailbox and message queues.
- Explain the concepts of latency, jitter, and response time.
- What is meant by a critical section?
- Explain the concept of priority inversion and how it can be mitigated / reduced?

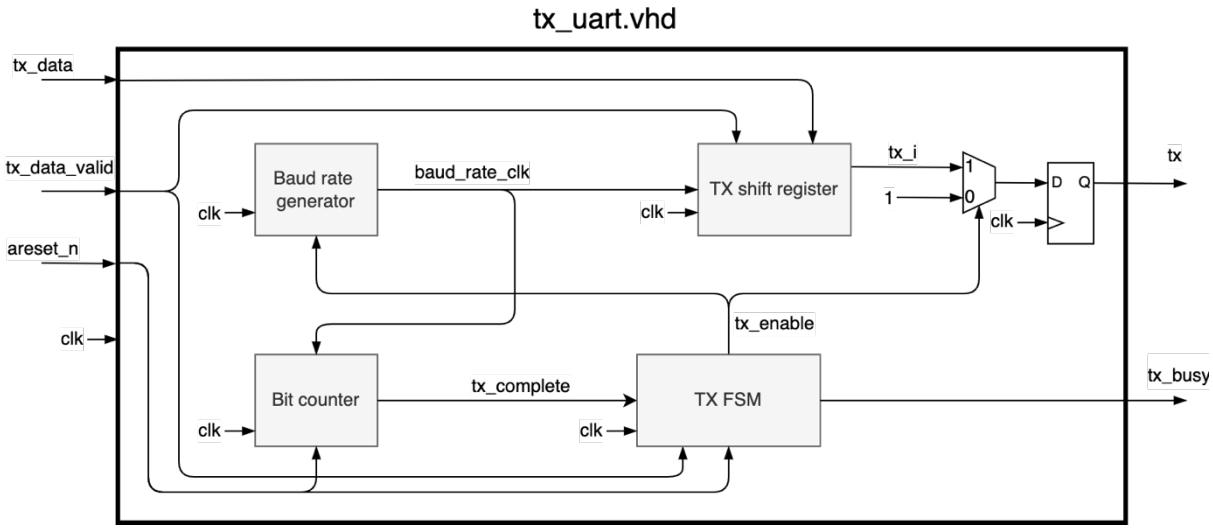
Version 1.3 - 16/11/2022

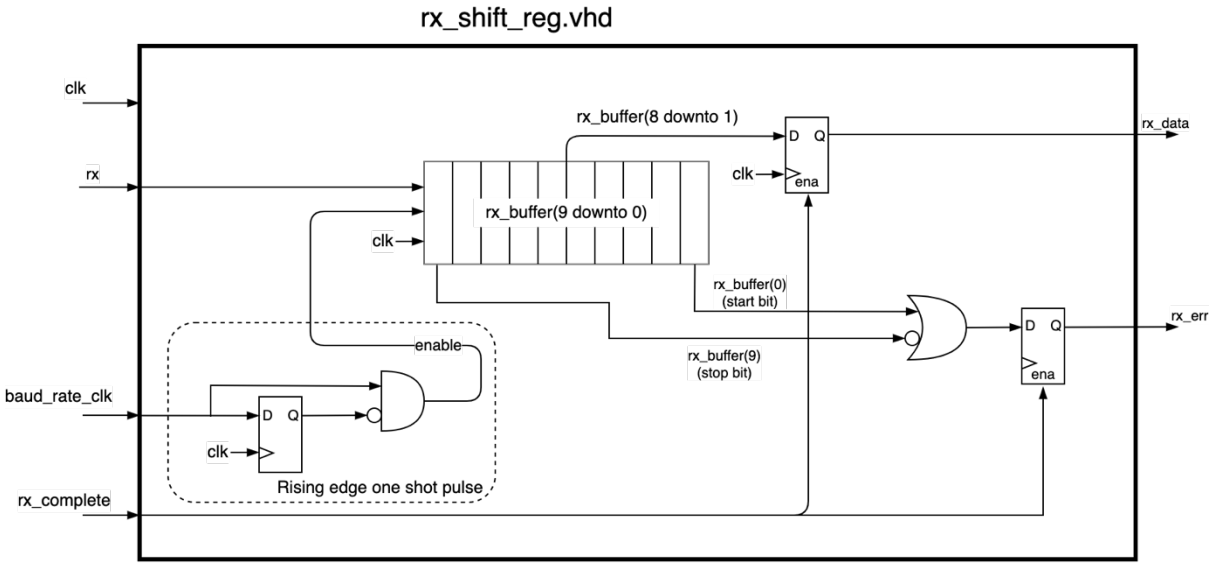
Some relevant figures

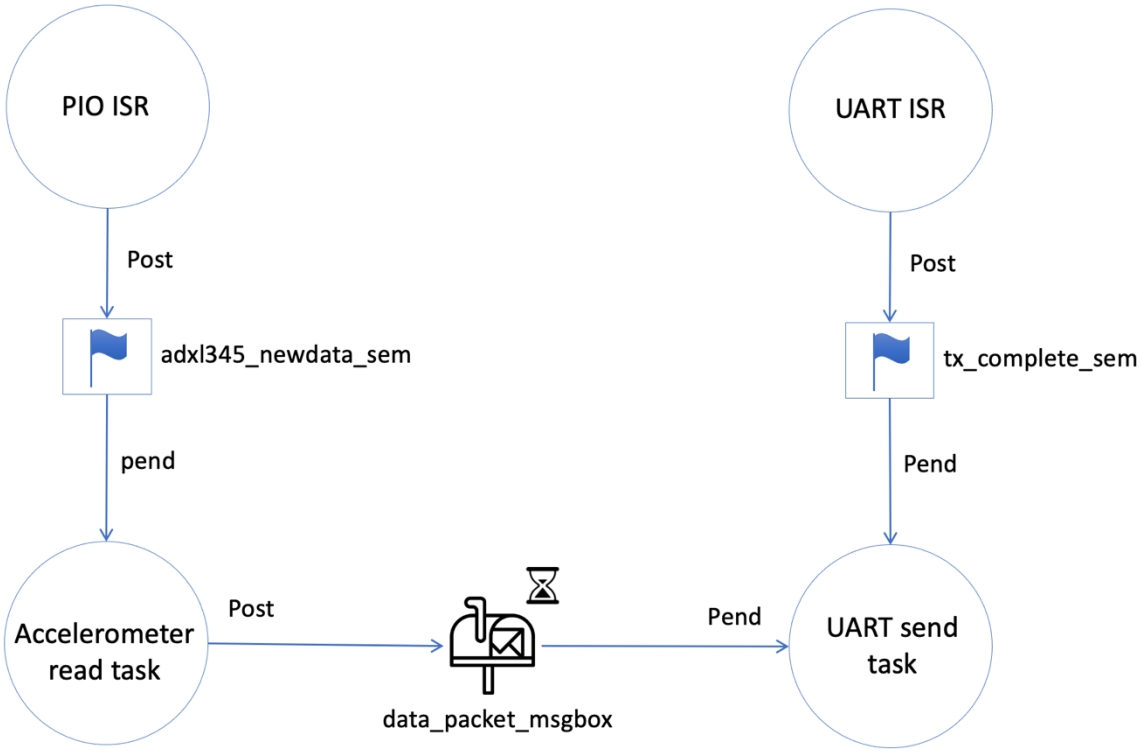





uart.vhd
tx_data
8
tx_data_valid
tx_busy
TX
UART TX
Processor bus
Processor interface
rx_data
8
rx_busy
rx_err
RX
UART RX


tx_uart.vhd
tx_data
tx_data_valid
areset_n
clk
Baud rate generator
clk
baud_rate_clk
TX shift register
clk
tx_i
1
1
0
clk
D
Q
tx
tx_enable
Bit counter
clk
tx_complete
TX FSM
clk
tx_busy


rx_shift_reg.vhd
clk
rx
baud_rate_clk
rx_complete
rx_buffer(8 downto 1)
rx_buffer(9 downto 0)
rx_buffer(0)
(start bit)
rx_buffer(9)
(stop bit)
enable
Rising edge one shot pulse
rx_data
rx_err
D Q
ena


PIO ISR
Post
adxl345_newdata_sem
pend
Accelerometer read task
Post
data_packet_msgbox
Pend
UART send task
UART ISR
Post
tx_complete_sem
Pend