Electronics System Design – End Semester Exam (April 2025)

Please note the following:

1. Please attempt 10 questions from the below set of questions.
2. Each question is for 4 marks. Total exam score is for 40 marks.
3. Total time for exam is 120 minutes.

Q1. Draw the System block diagram of the Embedded System made by you in the project ? Please explain different blocks with their interfaces and voltage levels. (CO3, CO4)

[ANS]

Marks distribution: [Each of 0.5 marks]

1. Input block shall be clear.
   a. It's interface to MCU to be clear
2. Analog or Digital (which type like GPIOs, I2C, SPI etc)
3. Output block shall be clear
4. Interfaces between different blocks shall be mentioned like GPIOs, I2C, SPI etc
5. Voltage levels of each blocks shall also be specified
6. Information about data storage can be important (if appliable)
7. Connection or information flow shall be clear [Each of 0.5 marks]
   a. From Input the arrow shall be going to MCU
   b. For both direction information flow: tow side arrow shall be made
   c. For outputs, the arrow shall be from MCU to output
8. Complete System block diagram (embedded plus environment like web..) showing how the project will be useful for user

Vague description shall be avoided or not given marks

****

Q2. Write down 5 technical challenges you encountered during the development of the project. How did you solve them? Explain with specifics. (CO2, CO4)

[ANS]
Marks distribution: [Each of 0.5 marks]
Following points are important:

1. Only technical points related challenges shall be mentioned

2. Logistics or unavailability of components related points shall be avoided.
    a. Unless otherwise it is explained how it was solved in detail
    b. No more than 1 point related to availability of the components or any other logistics shall be rated in assessment

    c. All technical points shall be explained in detail. Would be good to see challenges in the following:

        i. communication interface
        ii. voltage challenges
        iii. Interfacing
        iv. Coding challenges
        v. Tool chain issues
        vi. Pcb design issues
        vii. Other technical challenges

****

Q3. What are the different Clock sources available for STM32G0 ? What is the use of these sources? (CO3)

[ANS] Marks: 1 for each row. Names are important + Value Approx (0.5marks), usage 0.5marks

S.No	Clock Source	Name	Value	Usage and Comments
1	HSI	High Speed Internal Oscillator	16MHz	Internal RC oscillator used as the default clock source after reset. Suitable for general use without requiring external components.
2	HSE	High Speed External Oscillator	4 – 32 MHz	External crystal or oscillator. Used when higher accuracy is needed (e.g., USB, communication).
3	LSI	Low Speed Internal Oscillator	~32 kHz	Low-power internal RC oscillator. Typically used for the watchdog timer or RTC in low-power modes.
4	LSE	Low Speed External Oscillator	32.768 kHz	External low-speed crystal. Used for the Real-Time Clock (RTC) for precise timekeeping.

****

Q4. What are the decision making factors between the usage of a professional (license paid) or open-source tool chain that a product design engineer has to take. Justify with atleast 4 factors and examples in tabular form. (CO1,CO2)

[ANS] Each point is 1 mark. Comments are additional information

Factor	Professional Toolchain	Open- Source Toolchain	Comments
Cost & Budget	High upfront or subscription cost.	Typically, free to use	Cost can be a major factor for academic projects and startups. Product companies may use professional toolchains
Performance	High, faster compilation and linking.	Slower than professional.	Some software is designed for seamless integration with specific hardware or ecosystems, often with vendor backing. We must ensure the functions required and required performance benchmark and choose accordingly.
MCU Support	Toochain may support multiple MCUs from different vendors	Generally Provided by MCU provider.	Except some GCC or other toolchains which may support generic MCU cores.
Support and reliability	Comes with official support, updates, documentation and long-term maintenance	Community driven, may lack timely fixes or documentation.	For critical projects, timely and professional support is a must. One must consider reliable software with professional support.
Compatibility	Useful for any custom hardware	Useful and tested on General purpose hardware. May need some settings for custom hardware	
Compliance and Certification	Often compliant with industry standards (e.g. MISRA)	May lack formal certification or traceability	If the product needs to follow compliance or go through a certification, one must consider the tools that are compliant and certified, especially when dealing with safety-critical domains.
Configuration	Easy to configure	Difficult to configure	

****

Q5. Why do we see 32.768kHz XTAL on some of the evaluation boards and embedded designs ? Which applications would need them ? Give examples. Which applications will not require the usage of this XTAL, give examples. (CO1,CO3)

[ANS]
[Mark: 1]
The 32.768 kHz crystal oscillator (XTAL) is commonly seen on evaluation boards and embedded designs because it provides a highly accurate, low-power clock source especially suited for Real-Time Clock (RTC) functionality.

32768 can be divided by 2 (2^15) to reach the accuracy of 1Hz, which is 1 tick per second and hence useful for time-keeping. [1Mark]

Applications that Need the 32.768 kHz XTAL (1mark, 0.5 for RTC, 0.5 for examples)
These applications require accurate timekeeping, sleep/wake scheduling, or timestamping, especially during low-power operation:
1. Real-Time Clock (RTC) Applications
   Digital clocks, timers, alarms etc

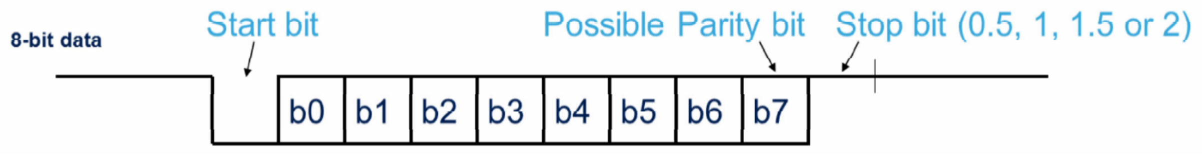
Additionally, Applications needing exact time keeping
For example: Energy Meter which needs to maintain exact calendar time or any medical device to record the exact time of reading.

Applications that Do Not Require the 32.768 kHz XTAL (1mark, 0.5 for examples)
These applications either don't need accurate time keeping and rely on internal oscillators
For example: Periodic applications like the following
1. Sending data time-to-time
2. Blinkers / Heart beat / LED blinking

****

Q6. Draw the UART data packet communication diagram between transmitter and receiver. Explain the bits of the communication packet. Explain with drawing. (CO3)

Diagram: (This is 2 marks. Important to show the Initial condition and START bit correctly)



Key points: (Each point is 0.5 mark)

a. Default condition of the Tx line is HIGH
b. Start bit: Falling low on the Tx line shows the start bit
c. Data bits are 7 or 8 or 9 depending on the settings used by user
d. Parity bit follows data bits: This is optional and as per Odd or Even parity setting selected by the user
e. STOP bit: HIGH on line cause the Stop bit. It is 0.5bit / 1bit or 2 bits

****

Q7. While making the prototype for a design, you have picked an Arduino shield which has digital IO devices. While interfacing it to STM32, what kind of challenges you foresee ? Explain with diagram. (CO1, CO2)

[ANS]

There are following issues: **[mark:1]**

1. Arduino Uno operates at 5V logic
2. STM32 operate at 3.3V

Challenge:

1. **Input from Arduino shield to STM32**
   High voltage Signal from Arduino shield can damage the STM32 GPIO due to high voltage. **[mark:1]**
   There are some 5V tolerant GPIOs available on STM32, which can tolerate the 5V signals. In such cases, the STM32 can operate without any issues **[mark:1]**

2. **Output from STM32 to digital input to Arduino**
   Signal swing of 3.3V from STM32 going to Arduino shield which is designed for 5V will check for **Vih inputs [mark:1].**
   Generally, Vih can be 5V * 0.7 = 3.5V. Since 3.3V from STM32 is still less than 3.5V input expected as high of Arduino, there is a possibility that signal is not recognized as High by Arduino. Hence the signal may not be recognized by MCU on Arduino **[mark:1].**

Solutions to Overcome Challenges **[mark:1].**

1. Using level shifters between 3.3V and 5V will help to interface
2. Using voltage divider can be used for Arduino 5V output to STM32 3.3V input.

****

Q8. For a Simplex mode communication, what is the difference between differential pair and single ended connection. Which one is better and why ? Draw the diagram. (CO3)

[ANS]
In simplex mode communication (one way only), the signal can be transmitted using either a single-ended or a differential pair connection.

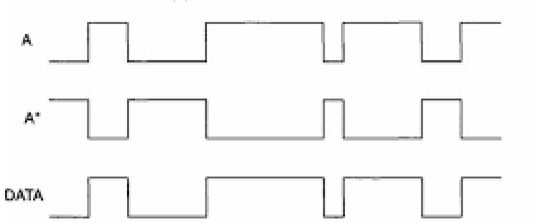
Single-ended connection: This method transmits an electrical signal over single wire, with the other wire connected to ground. It is simpler and cost-effective as it requires fewer wires. However, it is susceptible to EMI since the receiver references the signal to ground. The length of signal track or cable cannot be high because of issues related to noise etc. It is better for carrying data to short distances [Mark:1]

Differential pair connection: This approach uses two wires carrying signals of equal magnitude but opposite polarity. The receiver measures the voltage difference between the two signals, ignoring the common-mode noise. [Mark:1]

Differential pair connections are generally better in noisy environments or over long distances. Differential pair reject the common-mode noise and cancel out the electromagnetic interference. It also maintains signal quality over longer cables by mitigating voltage drops. However, a single-ended connection may be preferred in a low-cost, short -distance, or low-noise environment. [Mark:1]

Single Ended: [Tx]------->[Rx]

Differential Pair: [Tx+]--------->[Rx+]
[Tx-]--------->[Rx-]



****

Q9. For reading a serial memory at the required rate of 6Mbps, which kind of communication would you suggest to be implemented while designing digital interface for such a serial memory? Explain with reasons why this interface is better and not others ? (CO3)

[ANS]

Among the following serial peripherals: UART, SPI, I2C, let’s compare the features of each to see the suitability

1. UART is not used for Memory interfaces for MCUs because it is asynchronous interface. (Mark1)
2. I2C is useful for memory interface. However the I2C interface speed is 400KHz or maximum 1MHz. This would mean theoretical maximum can be 1Mbps (Mark1)
3. SPI is synchronous interface and can be used for high speed memory communication .
   For 6Mbps rate, SPI can be used. (Mark1)
4. There are further improvements done in SPI interfaces for high speed read/write which are:
   a. QSPI (Quad SPI) (Mark1)
   b. Octo SPI (Mark1)

****

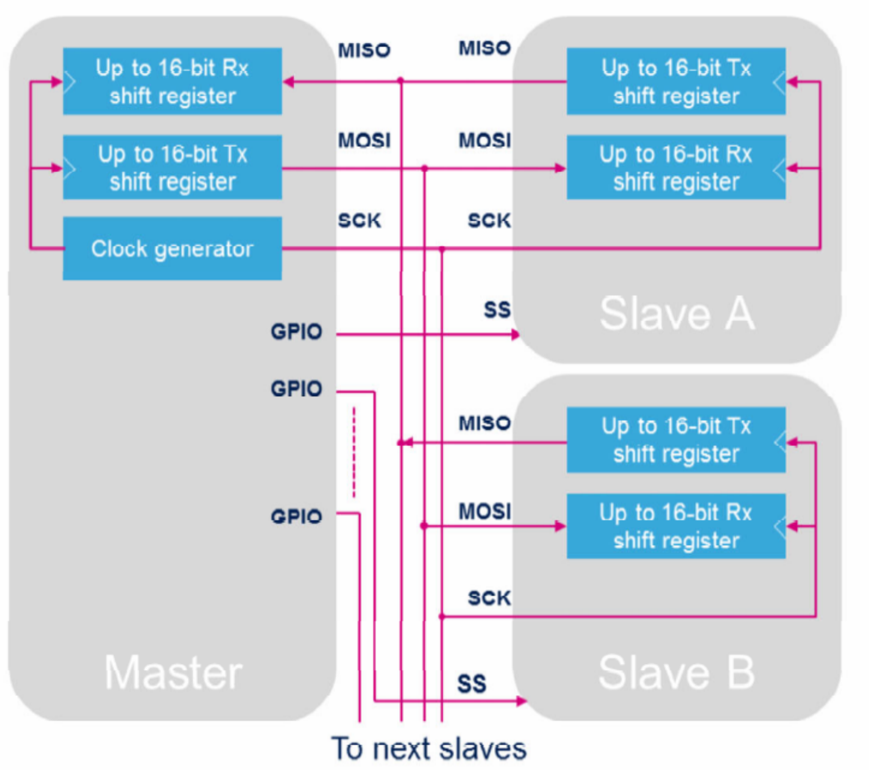
Q10. How to connect one master and 3 slave nodes using SPI communication ? Which considerations you will take for such interfacing. Make the diagram as well. (CO3)

[ANS]
The typical way for multi-slave SPI communication is the parallel (star) topology, which is used to link one master and three slave nodes

Consideration: (Marks: 0.5 for each point)

a. Every slave has a common MOSI, MISO, and SCLK line.
b. MOSI (Master Out Slave In): Connects from master to all slaves' MOSI pins.
   If configuration of MOSI pin is to be done, it can be PUSH-PULL
c. MISO (Master In Slave Out): Connects from all slaves' MISO pins to the master's MISO pin (shared line).
   If configuration of MISO pin is to be done, it can be Digital-Input (with or without PU)
d. SCLK (Serial Clock): Multiple GPIO connections from master to all slaves' SCLK pins.
   CLK pin is generally PUSH-PULL configuration
e. SS (Slave Select): Each slave gets a dedicated SS line from the master (SS1, SS2, SS3). Only one SS is LOW at a time to select the active slave
   SS pin is configured as PUSH-PULL output or Open-Drain with PU

Diagram: Important to have right pin names mentioned and arrows [Marks:2]



****

Q11. How does CSS: Clock Security System work in STM32 MCU. (CO3)
[ANS]
1)
The Clock Security System, CSS, in STM32 MCU is a fail-safe mechanism designed to monitor the external clock sources, such as the High-Speed External (HSE) an Low Speed External (LSE) oscillators and ensure system reliability by switching to a backup clock source in case of failure. [Marks=2]

For example, if CSS is used with HSE clock, it will continuously monitor the HSE signal. If a failure or fault is detected, CSS will trigger a clock security system interrupt and switch the system clock to the High-Speed Internal (HSI) RC oscillator and disable the HSE to prevent any further issues. [Marks=2]

The interrupt generated by CSS triggers a non-maskable interrupt, this executes indefinitely until CSS interrupt pending bit is clear. [Marks=1]

****

Q12. While configuring SPI peripheral of the STM32G0 or any other MCU, which all settings would needs to be configured ? (CO3)

[ANS]

Following settings need to be configured (mark:1 each)

1. Mode: Full-duplex mode
2. GPIOs to be configured MOSI, MISO, SPI CLK
   MOSI: Push-Pull
   MISO: Digital Input with or without PU
   SPI CLK: Push-pull
3. Slave select: GPIO output in PP Mode or OD with PU
4. Clock Polarity: CPOL
5. Clock Phase: CPHA
6. Data size
7. Sequence of data: MBb or LSb first

****