Question 1 [14 Marks]
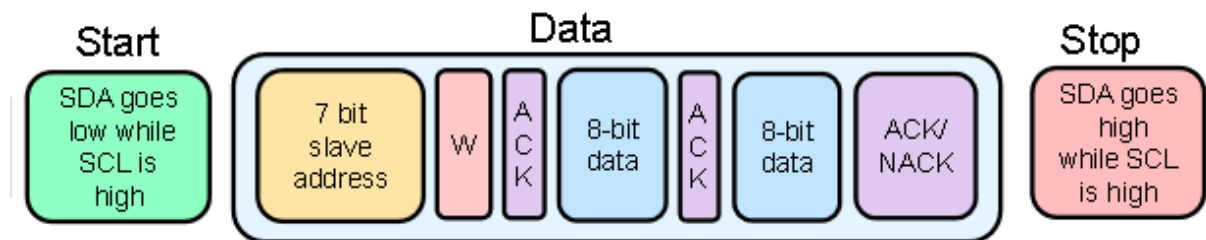
1.1. Give the message structure for I2C protocol when master communicates with slave. [4]
    1 Mark each:
- Start + stop CMD
- Address + data
- Read/write command
- Ack/Nacks

1.2. Give 2 advantages of I2C over SPI. [2]
    1 Mark each:
- I2C uses less hardware, 2 lines instead of 4 for SPI
- The message protocol includes acknowledge commands for verification purposes.
- Can support multiple masters.
- I2C is better for long distances

    If other, check validity before assigning marks.

1.3. Describe the start and stop conditions for I2C. [2]
    1 Mark each:
- Start: SDA goes low while SCL is high
- Stop: SDA goes high while SCL is high

1.4. Draw a timing diagram showing a Master sending 0b11010101 to slave at address 0b1110000. [6]
    1 Mark each unless specified:
- Start command; SDA goes low (1/2 mark)
- Correct slave address (1/2 mark)
- Write command; students can use "w" or "0" on SDA line
- Ack x2 (one after write command and one after 8-bit data); students can use "ack" or "0" on SDA line
- Correct data (1/2 mark)
- Labels (SCL and SDA)
- Showing that new data bits are essentially transmitted on the **falling** edge of the clock, and sampling occurs on the **rising** edges in the middle of each bit
- Stop command; SCL goes high and stops alternating, **then** SDA goes high at any point after SCL is high (1/2 mark)
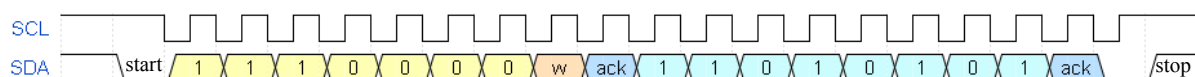
Question 2 [8 Marks]

2.1. Explain what Binary Coded Decimal is.

[2]

    1 Mark each:

- Binary-coded decimal is a system of writing numerals that assigns a four-digit binary code to each digit 0 through 9 in a decimal (base 10) number. Each digit of the base 10 number is represented individually using 4 bits. (OR SIMILAR)
- Example: 1764 -> 0001 0111 0110 0100

2.2. Mention one advantage and one disadvantage to using BCD. [2]

    1 Mark each:

    Advantages include:

- Allows to overcome size limitations on integer arithmetic.
- Easy conversion from machine to human readable numerals.
- Easy to code and decode (compared to binary).

    Disadvantages include:

- Requires more space in memory than binary.
- Increased circuit complexity.
- Wasteful (many unused 4-bit states)

    For others not in these lists, use Google and common sense.

Question 3 [3 Marks]

3.1. Explain briefly what Unix Epoch Time is and why it is used.
[2] 1 Mark each:
- In computing, Unix time (also known as Epoch time, Posix time, seconds since the Epoch, Unix timestamp or UNIX Epoch time) is a system for describing a point in time. It is the number of seconds that have elapsed since the Unix epoch, excluding leap seconds. The Unix epoch is 00:00:00 UTC on 1 January 1970.
- Unix time is a single signed number that increments every second, which makes it easier for computers to store and manipulate than conventional date systems. Interpreter programs can then convert it to a human-readable format.

3.2. What is the Epoch Time for 01 January 2023 at 12:00:00 AM (timezone GMT+02:00)? [1]

    1 Mark:
    Epoch = 1672524000

Question 4 [4 marks]

1. What is meant by a "Dynamic, Synchronous" Real-time System? [2]

   Student needs to mention that the arrival times of tasks are unpredictable (dynamic) and that tasks execute at a specific period (synchronous) in this kind of RTS. [1 mark each]

2. The main differences between RISC and CISC architectures were discussed in lectures. What are the two main discrepancies between these two in terms of instruction set size and memory access for operands? [2]

   RISC has a reduced instruction set size (around <100), whereas CISC has larger number of instructions (around >100 to 250). RISC can only operate on operands in registers, whereas CISC can operate on operands in memory and registers. [1 mark each]