A1 [25 Points].

[Part A]: To calculate IBRD (the integer part - the nearest integer will be used):

$$\text{IBRD} = \frac{\text{System Clock}}{\text{Desired Baud Rate} \times \text{Oversampling Rate}} = \frac{14,000,000}{115200 \times 16} \approx 7.59 \Rightarrow \text{IBRD} = 8$$

To calculate FBRD (the fractional part):

$$\text{FBRD} = \text{nearest integer of } \left(\frac{14,000,000}{115200 \times 16} \times 16 \right) \approx 10$$

[Part B]: To calculate the actual baud rate (based on the selected IBRD and FBRD):

$$\text{Baud Rate} = \frac{14,000,000}{16 \times (8 + \frac{10}{16})} \approx 101449 \,\text{baud}$$

The difference between this actual baud rate and the desired baud rate will determine the error rate:

$$\text{Error} = \frac{101449 - 115200}{115200} \times 100\% \approx -11.93\%$$

Since -11.93% exceeds the acceptable $\pm 2\%$ tolerance, the error is too high, likely leading to communication issues. So, oversampling of 16x is not the desired configuration for this system.

[Part C]: The other oversampling configurations must be tried to find the optimum oversampling ratio with the acceptable overhead. We start with oversampling of 8:
To calculate IBRD (the integer part):

$$\text{IBRD} = \frac{14,000,000}{115200 \times 8} \approx 15.2 \Rightarrow \text{IBRD} = 15$$

To calculate FBRD (the fractional part):

$$\text{FBRD} = \text{nearest integer of } \left(\frac{14,000,000}{115200 \times 8} \times 16 \right) \approx 3$$

To calculate the actual baud rate (based on the selected IBRD and FBRD):

$$\text{Baud Rate} = \frac{14,000,000}{8 \times (15 + \frac{3}{16})} \approx 115226 \,\text{baud}$$

The difference between this actual baud rate and the desired baud rate will determine the error rate:

$$\text{Error} = \frac{115226 - 115200}{115200} \times 100\% \approx 0.0228\%$$

With an error of 0.0228%, this configuration falls within the acceptable $\pm 2\%$ tolerance. Higher oversampling improves communication reliability, and as this configuration offers the highest oversampling ratio below 16x, it could be chosen as the optimal setting for this system.

A2 [20 Points].

[Part A]: The transmission time for the master to read the data includes:

- Slave address: 10 bits (2 bytes)
- Internal register address: 16 bits (2 bytes)
- Data: 24 bits (3 bytes)

The transmission flow would be:
START | **SL_ADDR+W** | **ACK** | **SL_ADDR2+W** | **ACK** | **REG_ADDR** | **ACK** | **REG2_ADDR** | **ACK** | **STOP**
START | **SL_ADDR+R** | **ACK** | **SL_ADDR2+R** | **ACK** | **DATA1** | **ACK** | **DATA2** | **ACK** | **DATA3** | **NACK** | **STOP**

For the first line (the first packet): $1 + 8 + 1 + 8 + 1 + 8 + 1 + 8 + 1 + 1 = 38 \; bits$

For the second line (the second packet): $1 + 8 + 1 + 8 + 1 + 8 + 1 + 8 + 1 + 8 + 1 + 1 = 47 \; bits$

With fast mode plus (1MHz): $(38 \; bits + 47 \; bits) \times 1 \mu s/bit = 85 \; \mu s$

[Part B]: As it is continuous between the master and the device located at 0x2AB, then stop bits can be dropped from the first packet. So,

$$(85\ bits - 1\ bits) \times 1\mu s/bit = 84\ \mu s$$

A3 [30 Points].

[Part A]: For given $V_{\text{in}} = 3V$,

$$\text{**Bit 5 (MSB, } C\text{):** Set to 1 — Effective DAC output} = \frac{5\text{V} \times C}{C + \frac{C}{3} + \frac{C}{9} + \frac{C}{27} + \frac{C}{81} + \frac{C}{243}} \approx 3.34\text{V}.$$

Since 3.34V > 3V, reset Bit 5 = 0.

$$\text{**Bit 4 (next, } \frac{C}{3}\text{):** Set to 1 — Effective DAC output} = \frac{5\text{V} \times \left(\frac{C}{3}\right)}{C + \frac{C}{3} + \frac{C}{9} + \frac{C}{27} + \frac{C}{81} + \frac{C}{243}} \approx 1.113\text{V}.$$

Since 1.112V < 3V, keep Bit 4 = 1.

$$\text{**Bit 3 (next, } \frac{C}{9}\text{):** Set to 1 — Effective DAC output} = \frac{5\text{V} \times \left(\frac{C}{3} + \frac{C}{9}\right)}{C + \frac{C}{3} + \frac{C}{9} + \frac{C}{27} + \frac{C}{81} + \frac{C}{243}} \approx 1.491\text{V}.$$

Since 1.491V < 3V, keep Bit 3 = 1.

$$\text{**Bit 2 (next, } \frac{C}{27}\text{):** Set to 1 — Effective DAC output} = \frac{5\text{V} \times \left(\frac{C}{3} + \frac{C}{9} + \frac{C}{27}\right)}{C + \frac{C}{3} + \frac{C}{9} + \frac{C}{27} + \frac{C}{81} + \frac{C}{243}} \approx 1.613\text{V}.$$

Since 1.613V < 3V, keep Bit 2 = 1.

$$\text{**Bit 1 (next, } \frac{C}{81}\text{):** Set to 1 — Effective DAC output} = \frac{5\text{V} \times \left(\frac{C}{3} + \frac{C}{9} + \frac{C}{27} + \frac{C}{81}\right)}{C + \frac{C}{3} + \frac{C}{9} + \frac{C}{27} + \frac{C}{81} + \frac{C}{243}} \approx 1.657\text{V}.$$

Since 1.657V < 3V, keep Bit 1 = 1.

$$\text{Effective DAC output} = \frac{5\text{V} \times \left(\frac{C}{3} + \frac{C}{9} + \frac{C}{27} + \frac{C}{81} + \frac{C}{243}\right)}{C + \frac{C}{3} + \frac{C}{9} + \frac{C}{27} + \frac{C}{81} + \frac{C}{243}} \approx 1.668\text{V}.$$

Since 1.668V < 3V, keep Bit 0 = 1.

****Final Digital Output:**** 011111 **(31 in decimal).**

[Part B]: For the resolution, the resolution of an n-bit ADC with reference voltage V_{ref} is given by (based on the smallest capacitance in the SAR network):

$$\text{Resolution} = \frac{5\text{ V} \times \left(\frac{C}{243}\right)}{C + \frac{C}{3} + \frac{C}{9} + \frac{C}{27} + \frac{C}{81} + \frac{C}{243}} \approx 0.0137366\text{ V per the smallest step.}$$

[Part C]: No, the voltage steps in this SAR ADC model are not uniformly increasing; instead, they vary irregularly due to the non-linear capacitive network $(C, \frac{C}{3}, \frac{C}{9}, \dots)$. For example, the voltage increment from 000000 to 000001 is 0.0137 V, while the increment from 000001 to 000010 is 0.0275 V, and from 000011 to 000100 it jumps to 0.0687 V. This non-uniform pattern results in inconsistent resolution across the ADC range, with some steps larger or smaller than others.

A4 [25 Points].

[Part A]: The following is the concatenation of all devices and the microcontroller:
μC, A, B, C = 5D, 3C, A7, E1

μC, A, B, C = 0101 1101, 0011 1100, 1010 0111, 1110 0001

After 5 clock cycles, the new data in each would be:
μC, A, B, C = 0000 1010, 1110 1001, 1110 0101, 0011 1111
μC, A, B, C = 0A, E9, E5, 3F

[Part B]: The following is the concatenation of all devices and the microcontroller:
μC, A, B, C = 5D, 3C, A7, E1
μC, A, B, C = 0101 1101, 0011 1100, 1010 0111, 1110 0001

After 11 clock cycles, the new data in each would be:
μC, A, B, C = 1111 1100, 0010 1011, 1010 0111, 1001 0100
μC, A, B, C = FC, 2B, A7, 94

[Part C]: Regarding the delay,

- With a clock speed of 1 MHz, each clock pulse takes $\frac{1}{1\,\text{MHz}} = 1\,\mu$s.
- To fully shift `0x5D` from Device A to Device C, it requires 8 bits $\times\, 3$ devices $= 24$ clock pulses.
- Total delay = 24 pulses $\times\, 1\,\mu$s $= 24\,\mu$s.