EECS 461, Winter 2023, Problem Set 7: SOLUTIONS[1]

1. Consider the CAN network in the lab, with 6 lab stations working in pairs to implement 3 haptic interfaces. Each lab station will read the encoder every T seconds, and write this value to the network. The partner lab station will take the encoder value, use it to compute a reaction torque, and transmit the reaction torque to the network. The torque value will be received by the original lab station, and used to update the PWM duty cycle that controls the motor. With 3 virtual walls implemented, it will thus be necessary to transmit 6 messages on the CAN bus every T seconds.

   Recall that a CAN message containing the maximum 64 bits of data will actually be 111 bits long due to overhead bits. Furthermore, with high speed CAN (500 kbit/sec) it takes $2\mu\text{s} = 0.002$ msec to transmit 1 bit on the CAN bus.

   If the effects of bit-stuffing are considered, there are additional bits that need to be added after every sequence of 5 bits of identical polarity, except in the CRC delimiter, ACK, EOF, and INT fields. Hence, there is a worst case maximum of $\lfloor \frac{34+\#\text{data bits}-1}{4} \rfloor$ stuff bits in the CAN message[2]. (The numerator of this formula follows because only 34 of the non-data bits are potentially eligible for bit-stuffing. The denominator is equal to 4 because each stuff bit may itself be the first bit of a 5 bit sequence.)

   (a) Suppose that all 6 CAN messages use the maximum 64 bits of data. How much time is required to transmit these messages? Consider the best case (no stuff bits) and worst case (maximum number of stuff bits) scenarios.

   SOLUTION: (no stuff bits) 6 messages $\times$ 111 bits/message $\times$ 0.002 msec/bit $= 1.332$ msec. (maximum stuff bits) $\lfloor \frac{34+64-1}{4} \rfloor = 24$ stuff bits $\Rightarrow$ messages with the maximum number of stuff bits are 135 bits long. Hence 6 messages $\times$ 135 bits/message $\times$ 0.002 msec/bit $= 1.62$ msec.

   (b) If the CAN message uses only 32 bits of data, how much time will it take to transmit all 6 messages? Consider the best case (no stuff bits) and worst case (maximum number of stuff bits) scenarios.

   SOLUTION: (no stuff bits) 6 messages $\times$ 79 bits/message $\times$ 0.002 msec/bit $= 0.948$ msec. (maximum stuff bits) $\lfloor \frac{34+32-1}{4} \rfloor = 16$ stuff bits $\Rightarrow$ messages with the maximum number of stuff bits are 95 bits long. Hence 6 messages $\times$ 95 bits/message $\times$ 0.002 msec/bit $= 1.14$ msec.

   (c) Again suppose that the 6 CAN messages use 32 bits of data. What is the minimum sample period T that will achieve 25% bus utilization assuming the worst case message length?

   SOLUTION: $1.14/T < 0.25 \Rightarrow T > 4.56$ msec. For an integer value, set $T = 5$ msec.

   (d) Suppose that the sample period must satisfy $T \leq 4$ msec in order to achieve a good virtual wall. What is the maximum number of virtual walls that can be implemented on the network with no more than a 25% bus utilization? Assume 32 bits of data per CAN message and the worst case message length.

   SOLUTION: We have seen that, with worst case messages, 3 walls require 5 msec. Hence we can implement at most 2 walls (which require 4 messages) under these conditions. Check: 4 messages $\times$ 95 bits/message $\times$ 0.002 msec/bit $= 0.76$ msec, and thus $0.76/4 = 0.19 < 0.25$, which satisfies the utilization constraint.

2. Low speed CAN (125 kbit/sec) is implemented with different hardware than is high speed CAN. In this problem, we will see how to configure our (high speed) FlexCAN module to run at 125 kbit/sec, and estimate CAN bus loading at this lower speed.

   (a) The CAN bit rate is given by the formula

$$\text{bit rate} = \frac{\text{CAN clock frequency}}{\#\text{ time quanta}},$$

   where

$$\text{CAN clock frequency} = \frac{\text{XOSC}}{\text{PRESDIV} + 1},$$

[1] Revised April 5, 2023.

[2] "$\lfloor \cdot \rfloor$" denotes the floor function: the largest integer less than or equal to the given argument

and

$$\#\ \text{time quanta} = \text{SYNC\_SEG} + (\text{PROPSEG} + \text{PSEG1} + 2) + (\text{PSEG2} + 1).$$

In these formulas, XOSC = 40 MHz, and SYNC_SEG = 1, and PRESDIV, PROPSEG, PSEG1, and PSEG2 are bitfields whose values are set in `flexcan.c`. Using the same values of PROPSEG, PSEG1, and PSEG2 as in Lab 7, find the value of PRESDIV that sets the CAN bit rate = 125 kbit/sec.

SOLUTION: According to `flexcan.c`, we have that PROPSEG = 2, PSEG1 = 5, and PSEG2 = 5. With these values and the fact that SYNC_SEG = 1, we have that # time quanta = 16. Hence

$$\text{bit rate} = \frac{40 \times 10^6}{16 \times (\text{PRESDIV} + 1)}.$$

To achieve a bit rate equal to 125 kbit/sec, we thus need

$$\text{PRESDIV} = \frac{40 \times 10^6}{16 \times 125 \times 10^3} - 1 = 19.$$

(b) Suppose that all 6 CAN messages use the maximum 64 bits of data. How much time is required to transmit these messages? Consider only the best case (no stuff bits) scenario.

SOLUTION: With a bit rate equal to 125 kbit/second, it takes 0.008 msec to transmit one bit. Hence 6 messages $\times$ 111 bits/message $\times$ 0.008 msec/bit = 5.328 msec.

(c) If the CAN message uses only 32 bits of data, how much time will it take to transmit all 6 messages? Consider only the best case scenario.

SOLUTION: With a bit rate equal to 125 kbit/second, it takes 0.008 msec to transmit one bit. Hence 6 messages $\times$ 79 bits/message $\times$ 0.008 msec/bit = 3.792 msec.

(d) Again suppose that the 6 CAN messages use 32 bits of data. What is the minimum sample period T that will achieve 25% bus utilization assuming no stuff bits?

SOLUTION: $3.792/T < 0.25 \Rightarrow T > 15.168$ msec. For an integer value, set $T = 16$ msec.

(e) Suppose that the sample period must satisfy $T < 4$ msec in order to achieve a good virtual wall. What is the maximum number of virtual walls that can be implemented on the network with no more than a 25% bus utilization? Assume 32 bits of data per CAN message and no stuff bits.

SOLUTION: Check two walls: 4 messages $\times$ 79 bits/message $\times$ 0.008 msec/bit = 2.528 msec, and thus $2.528/4 = 0.632 > 0.25$, which violates the utilization constraint. Check one wall: 2 messages $\times$ 79 bits/message $\times$ 0.008 msec/bit = 1.264 msec, and thus $1.264/4 = 0.316 > 0.25$, which also violates the utilization constraint. Hence, with low speed CAN, we cannot implement even one wall at a 4 msec update rate.

3. Write a Simulink model that will convert one 32 bit number into four 8 bit numbers using both Simulink blocks and an S-Function Builder block. Your model should appear as shown in Figure 1. You may wish to modify the S-Function Builder block for converting four 8 bit numbers into one 32 bit number that is found on the Canvas website. Hand in your Simulink diagram and a description of the changes you made in the S-Function Builder block.

SOLUTION:

There only a few changes that need to be made in the S-Function Builder block:

(i) the input `u` should have 1 row, and be of type `uint32`

(ii) the output y should have 4 rows, and be of type `uint8`

(iii) the block outputs should be computed as

```
unsigned int input_val = (unsigned int) u[0];
y[0] = (unsigned char)(input_val);
y[1] = (unsigned char)((input_val & 0x0000FF00) >> 8);
y[2] = (unsigned char)((input_val & 0x00FF0000) >> 16);
y[3] = (unsigned char)((input_val & 0xFF000000) >> 24);
```

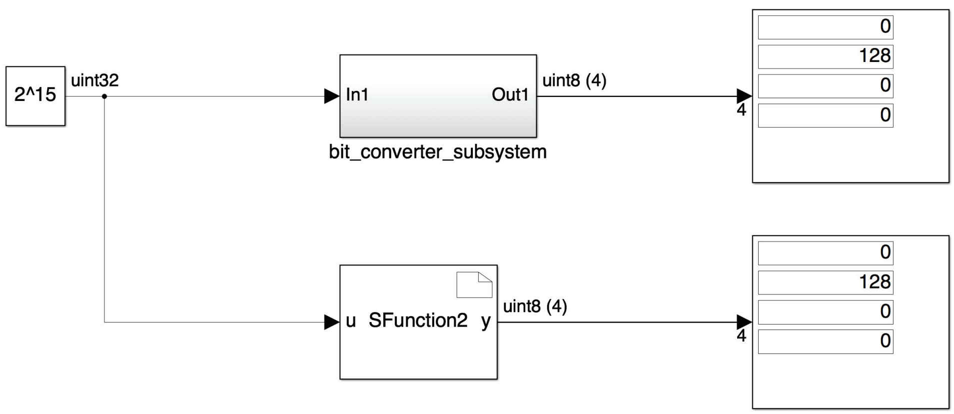

Figure 1: S-function to convert one 32 bit number to four 8 bit numbers.