CSE 4219– Principles of Embedded System Design

Assignment # 1
(Due: 03.11.2024 23:59)

For the following problems, write ARM assembly language programs. In the Keil MDK-ARM IDE, create a project for each, enter and build the program, and then execute and debug it in the Keil MDK-ARM debugger. You may run the programs in the simulation mode. You should write a minimum 2-pages long project report that describes how you implemented the project. In addition to explanations, for each program provide (1) a printout of the source program, and (2) one "screen capture" of the debugger memory window and register window, showing the program results at the conclusion of the program (The results written by the program). *Please circle the results in the memory window, register window or trace window.*

Q1. Write an assembly program that calculates the value of the following expression based on two input arguments *a* and *n*.

For example, when a= 3 and n = 5, we have

F(3_5) = 3 + 33 + 333 + 3333 + 33333

Please calculate the sum and keep it in r0.

Q2. In this question, you will implement an assembly program that swaps two columns in an integer matrix. First, you can initialize an input matrix with some initial values and a block of zeroed memory to store swapped result. Note that, two dimensional array should be stored as a single dimensional array in the memory, and the correct position of the corresponding columns should be calculated accordingly. You should keep two integers values that represents column indexes based on zero-based indexes. Assume that column indexes are valid.

Q3. In this question, you will implement a program that generates and stores Error Correcting Codes (ECC) for 8 bit input data. ECC is a widely used technique for detecting or correcting errors in data storage or data transmission. In order to correct one bit error and detect two bit errors in the data bits you can add 5 extra bits to stored or transmitted data.

Suppose that you have 8-bit data as the following

8-bit data: d8_d7_d6_d5_d4_d3_d2_d1

Then, you should expand it to 13-bit data by adding redundant bits as the following;

Expanded 13-bit data: d8_d7_d6_d5_p8_d4_d3_d2_p4_d1_p2_p1_p0

Here, p bits are known as *parity bits*.

Each parity bit checks certain data bits (according to the bit positions in the expended version) as shown in below:

- p1 is the even parity of bit positions 3, 5, 7, 9, 11
- p2 is the even parity of bit positions 3, 6, 7, 10, 11
- p4 is the even parity of bit positions 5, 6, 7, 12
- p8 is the even parity of bit positions 9, 10, 11,12
- p0 is the even parity of all bit positions 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12

In even parity, the number of bits with a value of 1 are counted. If the number of bits with a value of 1 is odd, parity bit arranged as 1; it is 0 if the count is even.

For example, if the given 8-bit data is the following:

1	0	1	1	0	0	1	1

The expanded 13-bit data is the following:

12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	1	1	0	0	1	0	1	0	1	1

Here the p1 is the even parity of bits positions 3, 5, 7, 9, 11, which makes 3 in total. The even parity of 3 is 1. So, p1 is 1.

Likewise, p2 is 0, p4 is 0, p8 is 1, p0 is 1.

You should keep 8-bit data in the memory region and zeroed block of 13-bit memory for the expanded data with parity bits. Please obey the calculation rules and stored bit positions as explained before.

Important Notes:

- You are required to work in groups of 3 people.
- *It should be noted that only selected question(s) will be graded.*
- We use tools that automatically detect plagiarism among the submissions!
- In case of any form of copying and cheating on solutions, you will get **FF** grade from the course! You should submit your own work. In case of any forms of cheating or copying, both giver and receiver are equally culpable and suffer equal penalties.
- Please zip ARM Keil project files and submit it using filename StudentNumbersPA1.zip (ex: 150713852_150713098_150101122_PA1.zip) to Canvas system (under Assignments tab).
- No late submission will be accepted.