Microprocessors lab ECSE426

Lab 1 Demo sheet

Group No:

Student Name:	Student ID:	Grade:	/6
Student Name:	Student ID:	Grade:	/6
Student Name:	Student ID:	Grade:	/6

1. **Demonstrate correct functionality of the code as follows:** **(1.5 marks)**
   - The outputs of the assembly Viterbi function matches the TA outputs for the three cases (0.6 mark)

   - The outputs of the C Viterbi function matches the TA outputs (0.2 mark)
   - The output of the C algorithm with thresholding matches the TA output (0.2 mark)

   - The assembly and C function prototypes match the ones listed in the experiment. The *struct* definition matches the one in the lab sheet. No dynamic memory allocation or unnecessary pointers. (One mistake loses all) (0.125 mark)

   - Assembly subroutine adheres to the calling convention and gives correct output if called from within C (0.125 mark)

   - Correct test benches and initialization subroutines (in assembly and C) (0.25 mark)

2. **Coding requirements and optimization** **(2.5 marks)**

   Grades awarded based on best results between groups. (Top 5 groups get highest mark, second top 5 groups get 75% of this part's mark ...etc.)

   Set simulation speed at 25MHz.

   - Execution time of assembly Viterbi function for one observation is (_____ µsec) (1 marks)
   - Execution time of C Viterbi function for one observation is (_____ µsec)

(0.75 marks)

- Efficient use of registers, number of Rx registers used (_____), number of Sx registers(_____) (0.5 mark)
- Efficient and correct use of either data or stack memory (0.25 mark)

3. C and assembly implementations discussion. Students should explain the code clearly and their instruction choices. Students must answer any question the TA asks related to project structure / Start-up files / Assembly functions / stack or data memory / calling convention / C-implementation / Assembly implementation **(Individual not group mark)**

- Student 1 _____ **(1.5 marks)**

- Student 2 _____ **(1.5 marks)**

- Student 3 _____ **(1.5 marks)**

4. Code documentation. Use of clear functional comments in both assembly and C **(0.5 mark)**

Bonus **(0.5 mark)**

1. Re-implementation of code in CMSIS-DSP and understanding the API and rationale behind using these functions (0.25 mark)

2. Correct output of the CMSIS DSP for the test vector that matches the non-DSP implementation (0.25 mark)