

COSE321 Computer Systems Design

Assignment #7

No late turn-in accepted

You want to see the benefit of using Thumb2, in terms of **code density** (that is, code size). So, you decide to write a C code and compile it with ARM instructions and Thumb2 instructions, respectively.

First, write a C program that sorts data in ascending order from the smallest to the largest. The input data is `indata[32]`. After executing your sorting program (any sorting algorithm is fine), the array `outdata[32]` should have the sorted integers of **all 32 input data** from the smallest to the largest. Note that you should add comments right next to each line of your code.

```
int indata[32] = { 2, 0, -7, -1, 3, 8, -4, 10,
                  -9, -16, 15, 13, 1, 4, -3, 14,
                  -8, -10, -15, 6, -13, -5, 9, 12,
                  -11, -14, -6, 11, 5, 7, -2, -12 };

int outdata[32];
```

Then, compile the C program using ARM instructions and Thumb2 instructions, respectively. Make sure the execution gives you the correct output.

- Compare the size of the compiled binaries for two cases (ARM vs Thumb2).
- When using Thumb2 instructions, how many (what percentage) of instructions are based on 16-bit and 32-bit?

What and How to submit:

1. Upload **your C program** to Blackboard.
2. Upload **video clip (2-min?)** to Blackboard. Your video clip should have **at least** the following contents:
 - Your smiling face
 - Understandable explanation of your C code
 - Demo on Zedboard with output memory dump
 - Code density & 16-bit and 32-bit instruction ratio discussion with Thumb2

Note: This is an individual assignment. You are welcome to discuss, but **DO NOT COPY** solutions. If you are found to copy solutions from others or slightly modify the solutions from others, both of you will be given 0 credits.