

Project: Part #7 for ECE 152

The Whole Enchilada

200 points

Must be submitted electronically by 10:00AM on Wednesday, April 23

IMPORTANT: Please get started early! That way, when problems arise (which they will!), you will have time to ask me and/or the TAs for help.

In this part of the project, you will demonstrate that you can use your pipelined processor to solve an interesting, non-trivial problem. You will write software that cracks a cryptographic code. We provide you with a description of the encryption and decryption algorithms (for a scaled-down version of TEA, the Tiny Encryption Algorithm), and you will write software that can take cyphertext (an encrypted message) and produce plaintext (the original message). As part of cracking the cypher, you will need to (exhaustively) search for the encryption key.

As part of this project, you will download your (possibly enhanced) processor design into the FPGA prototyping board, demonstrate that it works, and show how fast it can decypher the cyphertext we provide you.

1 Requirements

I have provided for you the C code that implements the encryption and decryption algorithms. It is at: <http://www.ee.duke.edu/~sorin/project/tiny.cpp>.

You will use the C code to understand how encryption and decryption work. Based on this understanding, you will write an assembly program (in the Duke 152/16 ISA) that decyphers a message that has been encrypted with an unknown key. We will give you three inputs: a plaintext message A, the encrypted cyphertext of message A, and encrypted cyphertext of message B. To find the key, you will exhaustively try all possible keys to see which enables you to decrypt cyphertext A. To make the runtime reasonable, you should assume that the key will not be larger than 2^{22} . Then, using this key, you will decypher

cyphertext B. These inputs should be put into the static memory of your program, such as in the following code snippet:

```
data:
cyphertextA:
    .word 123
    .word 456
plaintextA:
    .word 543
    .word 123
cyphertextB: // note that size(messageA) can differ from size(messageB)
    .word 12
    .word 32
    .etc
```

2 The Need For Speed

I will award bonus points to the three groups whose demos are the fastest (and correct!). 50 points for 1st place, 40 for 2nd, and 30 for 3rd. I also reserve the right to award 20 bonus points to any group that implements a particularly interesting feature, even though their demo is not one of the fastest three demos.

You are free to make the processor go faster using either or both of the following two approaches. You do not have to do any of this, though. You will get full credit if your processor correctly decyphers the message we provide you.

- Use the currently unused opcode in the ISA: I intentionally left one opcode unused. You can use that opcode for any purpose you want. You will want to slightly modify the assembler to recognize that opcode and generate appropriate code.
- Enhance the microarchitecture: If you want to add branch prediction, more pipe stages, a 2-wide pipeline (remember that midterm question?), or any other microarchitectural feature, you may do so.

Whatever you do, though, must still comply with the ISA specification. Remember: speed is great, but correctness is the most important feature.

3 Hardware Demos

As with the previous part of the project, we will have graded demos of the FPGA implementations that will be held in the ECE 154 lab (Hudson 202A) just after the submission deadline (exact times to be announced later). You will want to test your designs on the hardware long before then, to make sure they work correctly (even if they worked without problem in Quartus). To provide you with access to the lab with the prototyping boards (Hudson 202A), the TAs will hold office hours in the lab during the last week before the project is due.

4 Submission

Submit two files for this assignment: `enchilada.bdf` (the processor design) and `software.s` (the software).

You may re-submit as often as you like, but a re-submission will overwrite whatever you've previously submitted for this assignment. I will grade whatever has been submitted before 10:00AM on Wednesday, April 23.