“There’s More Than One Way To Do This” Lab

### Introduction

During this lab you will:

* Learn the characteristics of instruction set architectures (ISA).
* Identify the ISA characteristics of two modern architectures, x86 and ARM.
* Compare the Hack, x86, and ARM ISAs.

### Starters

Type your answers into the text box following each question. You may work with someone else on this lab, and submit one copy of this, including your answers, on Canvas.

*What are the names of everyone who worked on this lab with you, including yourself?*

|  |
| --- |
|  |

Using Internet resources (Wikipedia is fine), answer each of the following questions for the x86, ARM, and Hack architectures.

### Operand Size and Type

*What is the primary operand size? (This may vary, depending upon the architecture’s generation.) What operand type(s) is/are supported (integer, floating point, decimal, string, etc.)?*

|  |
| --- |
| x86: |
| ARM |
| Hack: |

### Instruction Encoding

*Are there a few or many instruction formats? Are instructions of a fixed or variable size? How many bytes per instruction?*

|  |
| --- |
| x86: |
| ARM |
| Hack: |

### Memory Addressing

*Can ALU instructions (add, and, etc.) access memory or do they only access registers? List several of the addressing modes (your list doesn’t need to be exhaustive). For each ISA, provide the definition of one of the addressing modes (select a different addressing mode for each ISA). What sizes of memory operands are supported (bit, byte, 16-bit word, etc.)?*

|  |
| --- |
| x86: |
| ARM |
| Hack: |

### Memory Model

1. *Does the ISA support a “flat” address space or are all addresses segment-based?*

|  |
| --- |
| 1. x86: |
| 1. ARM |
| 1. Hack: |

### Registers

1. *How many registers are available? Are they all general purpose, or do they serve various purposes? If they serve various purposes, identify them.*

|  |
| --- |
| 1. x86: |
| 1. ARM |
| 1. Hack: |

### Branching

1. *Do conditional branch instructions actually perform a comparison or do they rely upon condition codes set by other instructions?*

|  |
| --- |
| 1. x86: |
| 1. ARM |
| 1. Hack: |

### Endianess

1. *What is the ISA’s endianess?*

|  |
| --- |
| 1. x86: |
| 1. ARM |
| 1. Hack: |

### Classes of Instructions

*What types of instructions does the ISA support? For each ISA, list and explain one particularly unique instruction.*

|  |
| --- |
| x86: |
| ARM |
| Hack: |

### RISC vs. CISC

1. *Does the ISA fall into the RISC or CISC classification? Briefly justify your answer.*

|  |
| --- |
| 1. x86: |
| 1. ARM |
| 1. Hack: |