Drake Bridgewater & Ryan Phillips
April 7, 2014
CS 472 HW 1

(1.3) We said that the pattern of 1s and 0s used to represent an instruction in a computer has no intrinsic meaning. Why is this so and what is the implication of this statement?

As far as intrinsic meaning is concerned, this is the case with language in general. Information can only be communicated if both the sender and receiver can then encode/decode the words/sentences/messages/bits according to a shared schema. In the case of bit patterns the implication is simply that anything reading or writing these bit patterns must know the format ahead of time.

(1.5) Modify the algorithm used in this chapter to locate the longest run of non-consecutive characters in the string.

```
read first digit in the string and call it new_digit
set the current_run_value to new_digit
set the current_run_length to 1
set the max_run to 1

REPEAT
   read the next digit in the sequence
   IF new_digit != current_run_value
        current_run_length ++
   ELSE
        current_run_length = 1

   current_run_value = new_digit

   IF current_run_length > max_run
        max_run = current_run_length
UNTIL last digit is read
```

(1.8) What are the differences between RTL, machine language, assembly language, high-level language, and pseudocode?

RTL or Register Transfer Language is actual programming of the registers at gate level. This kind of programming can be done with Verilog, VHDL or actual drawing of the gates.

Machine Language is a set of instructions executed directly on the CPU. Each instruction performs one of usually 10 different operations.

Assembly Language is a lower level programming language for a computational system where the language is highly dependent on the architecture

High-level Language is a programming language with much abstraction from the details of a the computer. Many of the operations are either automated or a subset of functions within a single high-level language command.

Pseudocode is an informal, high-level description of the operating principle of a computer program.

(1.12) What is the difference between a computer's *architecture* and its *organization*?

Computer architecture implies structure and tells us something about how the components of the computer fit together and how the different hardware components of a computer system relate; whereas, **computer organization** is exactly how the circuits are laid out. The organization is vastly important in all aspects of computers but is can be seen as more important in dedicated devices; these dedicated devices such as a video decoder for a camera will orient the different electrical components in such a fashion to enable greater throughput for video related operations. [2]

(1.18) What is the von Neumann bottleneck?

"The Expression Von Neumann bottleneck has been coined to indicate that one of the limiting factors of the stored program computer is the path between the CPU and the memory." [1] The Von Neumann Bottleneck is caused when the limited throughput, data transfer rate, between the CPU and the memory compared to the amount of memory. This is partly because program memory and data memory cannot be accessed at the same time.

(1.33) Is Moore's law a law?

No, Moore's Law is not a law, as the book puts it this is an empirical observation, and people usually think of Moore's law as results of the current analysis which state that the number of components on an integrated circuit doubles every two years has been accompanied by a corresponding increase in the speed of integrated circuits.

References

[1] Alan Clements. *Computer Organization and Architecture.* Global Engineering: Christopher M. Shortt, themes and variations edition, 2014.

[2] Mohsin Shahzad. Difference between computer organization & computer architecture?, April 2011.