CS4110 - Project Assignment

Design and Implementation of an Insertion Sorting Algorihim

Candidate: 8505

Candidate: …

Candidate: …

Candidate: …

*Abstract* — The scope of this document discusses the consideration of design and implementation of an insertion sorting algorithm in both hardware and software. We will go into greater detail with our insight and present the four implementations for our insertion sorting algorithm that consists of an ASIP, an FSMD, a single-thread C program in Zynq/Vitis IDE, and an IP generated with Vitis HLS and used in Zynq/Vitis IDE.

Keywords—Hardware, Software, Basys-3, Zybq, RTL, HLS, C, VHDL, FSMD, ASIP ISA, Vivado, Vitis

# Introduction

This assignment is written as a technical paper that goes through our process regarding our design and implementation of an insertion sorting algorithm with an input array of 8-bit unsigned numbers for four tasks in mind: 1) expanding our basic ASIP ISA, 2) creating an FSMD in VHDL, 3) coding it in Vitis IDE / Zynq and 4) creating a user IP block in Vitis HLS.

The AMD's Vitis Core Development Kit with the recommended version of 2023.2 was needed for this assignment as it includes applications like Vivado and Vitis HLS for our hardware and software implementation respectively.

Two development boards were used to carry out our testing and implementation of our insertion sorting algorithm. This includes the Digilent Basys-3 [2] and the Digilent Zybq-7000 (or Zybo-Z7) [3]. The Basys-3 was utilized for our first and second task because it was equipped with a Field Programmable Gate Array (FPGA) [2]. The ZYBO board was used for the third and last task as ZYBO development board has a Zynq 7000 System on Chip (SoC), which includes an FPGA, an ARM Cortex-A9 CPU, and connectivity peripherals [3].

# Insertion Sorting Algorihim

An insertion sorting algorithm is a very simple, straightforward method for sorting entries in a list or array in an ascending and descending order. The algorithm constructs a sorted segment of the array of one element at a time, placing each element into its proper position relative to the already-sorted area [1].

Initially, the algorithm treats the first member of the list as a sorted sublist with only that one entry. It then considers the next element in the list and compares it to those in the sorted sublist. If the new element is smaller than any of the elements in the sorted sublist, it is relocated until it reaches its proper position. This procedure repeats for each successive entry in the list, with the algorithm ensuring that the sorted sublist expands one element at a time while keeping its order.

(MAYBE SHOW PSEUDO CODE)

In the case of a list of numbers such as [4, 3, 5, 1, 2], insertion sort would begin at the first element of the array by considering element (4) as a sorted sublist. After comparing element 3 with element 4, it determines that element 3 should come before element 4. 3 is inserted in front of it, while 4 is moved to the right. The third element (5) is then considered by the algorithm. The two sorted elements (3 and 4) are already less than 5, thus nothing is changed. After the other elements in the sorted sublist (3, 4, and 5) have been shifted, the fourth element (1) is inserted at the front after being compared with every other element. The fifth element (2) is finally added between 1 and 3, resulting in the fully sorted list [1, 2, 3, 4, 5].

(SHOW FIGURE)

Insertion sort has the advantage of sorting the list in place, which eliminates the need for additional memory for a second list. This increases efficiency in terms of space complexity. However, it has a somewhat high time complexity for big lists, as it must constantly compare and shift entries, resulting in a worst-case time complexity of O(n²), where "n" is the number of elements in the list.

(SHOW GRAPH OF TIME-COMPLEXITY)

# ASIP

(WRITE SOMETHING HERE…)

* Assembly code and using the instruction from a scimatics (one cycle cpu)
* Incorprate it in Vivado as VHDL

# FSMD (Claire)

(WRITE SOMETHING HERE…)

* Creating a scimatics before coding
* Coding and debugging.

# Single-Thread C-Program (Claire)

(WRITE SOMETHING HERE…)

* Used Vivado to generate a IP block and wrapper
* Used IP block to generate a platform on Vitis
* Created a single-thread c program that insertion sorted elements in an array

# HLS Synthisis

(WRITE SOMETHING HERE…)

* Creating an HLS platform in Vities and C files (c file and header)
* Creating test bench
* Check if it works through c simulation
* C synthesis it
* Run ‘Package’
* Open Vivado and create an IP block (ZYNQ)
* Add IP-block from package and connect with ZYNQ
* Wrap it
* Bitstream and export as hardware (include bitstream)
* Use the exported file and add it as platform
* Build platform
* Create application
* Adjust main
* Run putty
* ez

# Using the Template

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

## Authors and Affiliations

**The template is designed for, but not limited to, six authors.** A minimum of one author is required for all conference articles. Author names should be listed starting from left to right and then moving down to the next line. This is the author sequence that will be used in future citations and by indexing services. Names should not be listed in columns nor group by affiliation. Please keep your affiliations as succinct as possible (for example, do not differentiate among departments of the same organization).

### For papers with more than six authors: Add author names horizontally, moving to a third row if needed for more than 8 authors.

### For papers with less than six authors: To change the default, adjust the template as follows.

#### Selection: Highlight all author and affiliation lines.

#### Change number of columns: Select the Columns icon from the MS Word Standard toolbar and then select the correct number of columns from the selection palette.

#### Deletion: Delete the author and affiliation lines for the extra authors.

## Identify the Headings

Headings, or heads, are organizational devices that guide the reader through your paper. There are two types: component heads and text heads.

Component heads identify the different components of your paper and are not topically subordinate to each other. Examples include Acknowledgments and References and, for these, the correct style to use is “Heading 5”. Use “figure caption” for your Figure captions, and “table head” for your table title. Run-in heads, such as “Abstract”, will require you to apply a style (in this case, italic) in addition to the style provided by the drop down menu to differentiate the head from the text.

Text heads organize the topics on a relational, hierarchical basis. For example, the paper title is the primary text head because all subsequent material relates and elaborates on this one topic. If there are two or more sub-topics, the next level head (uppercase Roman numerals) should be used and, conversely, if there are not at least two sub-topics, then no subheads should be introduced. Styles named “Heading 1”, “Heading 2”, “Heading 3”, and “Heading 4” are prescribed.

## Figures and Tables

#### Positioning Figures and Tables: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation “Fig. 1”, even at the beginning of a sentence.

1. Table Type Styles

| Table Head | Table Column Head | | |
| --- | --- | --- | --- |
| Table column subhead | Subhead | Subhead |
| copy | More table copya |  |  |

1. Sample of a Table footnote. (*Table footnote*)
2. Example of a figure caption. (*figure caption*)

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader. As an example, write the quantity “Magnetization”, or “Magnetization, M”, not just “M”. If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write “Magnetization (A/m)” or “Magnetization {A[m(1)]}”, not just “A/m”. Do not label axes with a ratio of quantities and units. For example, write “Temperature (K)”, not “Temperature/K”.

##### Acknowledgment *(Heading 5)*

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression “one of us (R. B. G.) thanks ...”. Instead, try “R. B. G. thanks...”. Put sponsor acknowledgments in the unnumbered footnote on the first page.

##### References

1. Wikipedia Contributors, “Insertion sort,” Wikipedia, Sep. 07, 2024. https://en.wikipedia.org/wiki/Insertion\_sort# (accessed Oct. 20, 2024).
2. Digilent, “Basys 3 FPGA Board Reference Manual”, April 2016.
3. Digilent, “ZYBO FPGA Board Reference Manual”, February 2017.