Advanced Embedded Systems

Project Topics Spring 2018-2019

ES_1, ES_2

Approximate Computing targeting improved performance, lower power consumption

In use cases such as matrix operations (multiplication, inversion, etc.), equation solving, image filtering (e.g., sobel, canny, prewitt), etc.

a) Software-only techniques, e.g.,

https://www.mentor.com/hls-lp/downloads/ac-datatypes

https://hlslibs.org/#collapseACDatatypes

https://github.com/hlslibs/ac_types

https://github.com/minhhn2910/fpPrecisionTuning

b) Using Vivado HLS tools, e.g.,

https://github.com/Xilinx/HLx_Examples

ES_3,4,5

Exploration of specific use cases on ZedBoard in terms of performance based on:

a) Using cache or Scratchpad Memory (on-chip SRAM)
b) Similar to (a) while taking advantage of DMA controller
c) Developing Baremetal apps and/or FreeRTOS

https://github.com/AVINASH793/String-matching-algorithms-and-applications

http://dis.unal.edu.co/~fgonza/courses/2006-II/algoritmia/approx_string_matching.pdf

ES_6,7

Exploration of Vivado HLS, by modifying the AverageCalculator example to

a) Compute different functions
b) Use multiple hardware cores managed by software

ES_8

Embedded Systems Project (Zedboard-based)

The objective is to develop an application in Baremetal using both cores of Zedboard, the A9_0 for control/monitor functions and A9_1 as the worker core.

A. (70%)

The functionality of each core application is listed next:

- A9_0 initializes data array0
- A9_0 notifies A9_1 to process these data
- A9_0 monitors A9_1 to identify completion of processing
- A9_0 while monitoring A9_1 prepares data array1
- A9_1 waits for signal from A9_0 to start searching a chunk of data: lines of text with varying length (and a pointer to the data array, and number of lines)

Push buttons:

a. TOP: instructs A9_0 to start processing (search for your NAME) in the first chunk of data (10 lines), output (on the OLED Display): line and column that was first found, number of occurrences

b. BOTTOM: time in cc that (a) spent for the 1st found, for the total N of occurrences

c. LEFT: as (a) but for 1000 lines

B. (30%)

Use the A9_1 to perform approximate string matching for sequence alignment

---

Important Notices:

You may reuse code found in the web, such as:

i. Boyer-Moore string search algorithm implementation in C
https://github.com/likejazz/boyer-moore-string-search

ii. Tom Van Court and Martin C. Herbordt,
"Families of FPGA-Based Accelerators for Approximate String Matching"
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3096528/

C implementations of Needleman-Wunsch and Smith-Waterman sequence alignment
https://github.com/noporpoise/seq-align
https://github.com/skotchandsoda/needleman-wunsch
http://www.rolfmuertter.com/code/nw.php