Module Code
COM00003H

BSc, BEng and MEng Degree Examinations 2022–23

DEPARTMENT OF COMPUTER SCIENCE

Embedded Systems Design and Implementation

Second Lab Assessment

Open Individual Assessment

Issued: March 15th 2023

Submission due: May 10th 2023, 12:00 noon

Feedback and marks due: June 7th 2023

All students should submit their answers through the electronic submission system: http://www.cs.york.ac.uk/student/assessment/submit/ by May 10th 2023, 12:00 noon. An assessment that has been submitted after this deadline will be marked initially as if it had been handed in on time, but the Board of Examiners will normally apply a lateness penalty.

Your attention is drawn to the section about Academic Misconduct in your Departmental Handbook: https://www.cs.york.ac.uk/student/handbook/.

Any queries on this assessment should be addressed by email to Ian Gray at ian.gray@york.ac.uk. Answers that apply to all students will be posted on the VLE.

Your exam number should be on the front cover of your assessment. You should not be otherwise identified anywhere on your submission.

Overview

In this assessment you will implement and accelerate a solver for a tiling puzzle. Your system will communicate with both the user and a network server to receive problems to solve. You will visualise the problem using graphics and will attempt to use the parallelism inherent in the FPGA to accelerate your design. You will describe your approach and design in a report.

For marking, the assessment is split into:

- [40 marks] Written report describing your key design, implementation, evaluation and testing activities.
- [60 marks] Evaluation of your developed system.

Further details of the report and evaluation are given later in this exam paper.

Problem

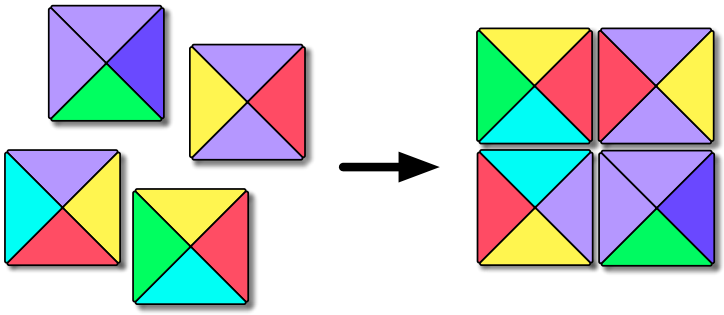

Figure 1: Example puzzle

Each puzzle consists of a set of square tiles and a grid. Each tile has four coloured edges. The puzzle involves placing the tiles onto the grid so that all touching edges are the same colour. Each tile may only be used once, and tiles may be rotated (but not mirrored). There are multiple solutions to a puzzle, and all puzzles have a solution.

Colours will be represented by numerical values and may have up to ten different values. You will need to visualise the puzzle. You can use any arrangement of colours or patterns, but the ten values must be easily distinguishable.

There will be a puzzle generator on the network that your solution will request puzzles from.

Required Functionality

Your system must do the following, all with one bitfile and application program. You can reprogram the FPGA during the demonstration, but only with the same bitfile and executable each time.

- Send an Ethernet message to the puzzle generator asking for a puzzle. Your solution should be able to request different puzzles based on input from the user. Technical details (the UDP packet formats, the protocol, and the IP address and port of the server) are on the assessment information page:
  https://wiki.york.ac.uk/display/RTS/EMBS+2023+Assessment+Information
  - Puzzle sizes may range from 2x2 to 8x8.
  - Puzzles have a 4-byte random seed. The user must be able to type in the seed they wish.
- After requesting it, receive the puzzle from the generator.
- Solve the puzzle, finding multiple solutions.
- Visualise the solutions graphically. You can draw the puzzle however you wish, as long as it is clearly understandable.
- If the user sends `n` (next) or `p` (previous) over the serial, the system should display different solutions. Some puzzles have literally millions of possible solutions, so you will not be able to store all of them! In this case, store a "reasonable" amount for display.
- If the user presses `a` (abort) whilst your system is searching, it should abort the search and display any solutions found so far.

You can solve the puzzle any way that you like, but you will have to use some kind of search algorithm. You must use custom Vitis HLS components to accelerate the search as much as possible. Note that if you do not use HLS and only have a software solution then the maximum amount of marks that you can get will be limited, because you will detail in your report the acceleration strategy you used and how effective it was.

Graphics

You should output the solved puzzle graphically. Information and example code for this can be found from the assessment information page:

You can draw the puzzle any way you wish, however it must be possible to clearly differentiate between tiles and the ten colours. If you have a visual impairment that causes this to be difficult, please speak to the module leader, or `cs-disability@york.ac.uk`.

Tips

You will be asked to *justify* your use of hardware parallelism. This is a parallel problem, and parallelism may be implemented in a variety of ways.

The report discusses the use of parallelism and the split of hardware/software functionality. Accordingly, your time would be better spent creating a simple software solution and then getting it working on the hardware, rather than optimising a clever software algorithm which is then difficult to implement in HLS at all.

You might find that your solution is “fast enough” in software alone. You should still focus on collecting data which demonstrates how a hardware implementation can be used to improve execution times so that it could handle more complex scenarios etc.

Your report asks for evidence of evaluation and testing. While working, consider collecting information on design size and execution times, so that you can talk about what you did, what effect it had, and how you know that you created a good solution.

Constraints

Your solution:

- must use the EMBS FPGA hardware platform.
- must be created using Xilinx tools, Xilinx IP cores, and your own custom hardware peripherals built using Vitis HLS.
- should **NOT** use other IP cores or designs available from the web or elsewhere.

Marks, Notes and Guidance

Written Report [40 marks]:

- Maximum of 1500 words - not including figures, tables, references, code fragments etc.
- Your report should assume that the reader is knowledgeable about embedded systems, the development toolkits (e.g. Vitis HLS), and the examination FPGA toolkit.
- Your report will discuss:
  - [5 marks] Design of system, including justification of the split of functionality between CPU and accelerator
    Criteria: clarity and completeness of discussion; quantitative evidence (execution times, hardware size, etc.) when justifying split of functionality between CPU and custom hardware.
  - [15 marks] Parallelism used within the system
    Criteria: clarity and completeness of discussion. Quantitative evidence for effectiveness of both design decisions and directives. Quantitative evidence to show effectiveness of custom hardware for accelerating simulation.
  - [15 marks] Evaluation and Testing
    Criteria: completeness of evaluation; discussion and justification of hardware resources used; discussion of timing performance of the system including breakdown of timing bottlenecks and discussion of how these would be avoided/improved. Discussion of testing strategy.
  - [5 marks] Overall quality
    Criteria: effective use of formatting, illustrations, quality of writing, etc.

Note that the 1500 word limit is a *maximum*, not a target.

Evaluation by Demonstration [60 marks]:
You will demonstrate your solution to an examiner. You will demonstrate your solution against a randomly selected set of puzzles to show its functionality.

Partial marks can be awarded where appropriate, for example if a small bug causes a feature to be unstable then some credit could still be awarded.

- [5 marks] Request specified puzzles from the server as instructed.
- [8 marks] Solve a 4x4 puzzle in < 1 sec and display at least one solution (you may output via the serial if graphics is not supported).
- [8 marks] As above but for a 6x6 puzzle
- [8 marks] Solve an 8x8 puzzle in < 5 seconds.
- [5 marks] Display a puzzle solution of at least size 4x4 graphically (may be a hard-coded solution if required).
- [6 marks] Display at least 8 possible solutions to a puzzle, and use the serial to switch backwards and forwards between them.
- [20 marks] Your design will use hardware to accelerate the software. Discuss and show your code for how you used parallelism and HLS to accelerate the search.

The markscheme for the final point is:

1. (0 marks) - Your design is purely a software-based solution.
2. (10 marks) - Correct use of an IP core to help solve the puzzle.
3. (15 marks) - Evidence of functional parallelism in the hardware implementation.
4. (20 marks) - Use of multiple IP cores, or similar advanced parallelisation techniques.

Remember that the use of hardware acceleration and offloading computation to the FPGA logic will be the main subject you will discuss in your report. Even if your solution is “quick enough” you should carefully consider what functionality to offload to the FPGA, and which to leave in the ARM core, and how to make the solution faster using parallelism.

Source Code
All code should be submitted electronically in a single archive (zip) file named Yxxxxxxx.zip where xxxxxxx is your assessment Y-number.

You should include in the archive:

- Your entire Vivado project (just include the entire folder)
- Your Vitis workspace. *Please try to remove any unneeded applications/systems etc. first* so that we ensure we are checking the correct code.
- Your HLS project (just include the entire folder)
- Your report in PDF, Word, Pages, or RTF format.

Be careful that you don't run out of diskspace when creating the archive. If in doubt, create the archive in `/tmp/` before uploading it to the submission point. Always download your submission back from the submission point, unzip it, and check its contents.

End of examination paper