DAT480: Reconfigurable Computing
Exam - January 2023

Date: January 9, 2023

Time: **8:00-12:00**

Examiner: Ioannis Sourdis

Department: Computer Science and Engineering

Inquiries: contact through phone, phone extension 1744

Duration: 4 hours

Grading scale: 100 points in total

Chalmers:
0: 0%-49%, 3: 50%-64%, 4: 65%-84%, 5: 85%-100%
GU:
Fail (U): 0%-49%, Pass (G): 50%-79%, Pass with Distinction (VG): 80%-100%

Available references: Lecture notes are allowed.
Textbooks or Calculators are not allowed.

General: Submit your solutions, in English, on blank paper sheets. Write legibly; feel free to use figures to get your point across.

The order of answering the questions does not matter (start with the easiest ones).

Please start the solutions for each problem on a new sheet. Please number the sheets so that the solutions are in numerical order.

Note that it is possible to receive partial credit for an answer even if it is not 100% correct.

Your personal identity code is required on each submitted sheet!

Good luck!

Note: *These are example answers to the exam questions for one possible set of randomized variables or explanations as of how a solution to an exam question can be derived.*

Question 1 Reconfigurable Devices: (15 points)

a) Draw the block diagram of a 4-input logic cell including its input and output connections to the nearby wires (connection to the reconfigurable interconnects). How many bits of configuration in total are needed? (7.5 points)

b) In the past years, newer CMOS technologies improved the latency of the logic (switching speed of transistors) more than the latency of the wires. As a consequence, logic has been getting relatively faster compared to the wires and the gap between speed of logic speed of wires has been growing. Assuming this trend continues, would you choose larger or smaller LUTs for future FPGAs and why? (7.5 points)

Answer

a) Similar to lecture on Reconfigurable Devices, slides 17 and 32. Configuration bits: it depends on how many wires you connect the logic cell to. If following the slide 32 then 17 for the logic cell plus 11 for the connections if input and output, so 28 configuration bits in total.

b) If the technology trend is that the wires get relatively slower than the logic, then it is better to need fewer wires for the implementation of a design (each combinational path in the design would be faster if it has fewer wires). Larger LUTs means that the implementation would need fewer LUTs. Fewer LUTs means fewer wires are needed to interconnect them (which is what we need). So, for the delay of our implementation it is better to use larger LUTs in newer technology nodes.
This can be also observed slide 27 in lecture on Reconfigurable Devices where it is shown that higher interconnect delay pushes the choice of LUT to a larger number of inputs.

Question 2 Compute Models & Applications: (15 points)

a) Select one dataflow or sequential-control compute model, from the ones you have seen in the course, and describe the pattern matching problem of your lab project having in mind to use an FPGA as an implementation platform.

b) Does the compute model you chose belong to the dataflow family or to the sequential control family? Which particular variant did you choose and why?

c) How does your choice affect: (i) capturing parallelism (what types of parallelism can you capture) and performance, (ii) determinism, and (iii) expressiveness compared to the other family of compute models.

Answer

(other answers can also be correct if justified properly).

a) I would choose a Dynamic Streaming Data Flow described as a basic CAM or DCAM shown in lecture on Applications, e.g. slides 26-27 (basic CAM).

b)
   i. Why Dataflow: to capture parallelism of (1) incoming streaming data, and (2) matching multiple patterns in See lecture on Applications, slides 26-27 (basic CAM).
   ii. Why dynamic streaming: to be able to "select" one of multiple concurrent matches (with a priority encoder). (one may decide to go for a "Dynamic Streaming Data Flow with Peeks" to allow merging of multiple concurrent matches but then you lose determinism and cannot guarantee a fixed processing throughput)

c)
   i. I can capture data parallelism (by pipelining) and operation/task parallelism (matching different patterns) this gives me high performance. Data parallelism would be more difficult to be captured in sequential control due to the streaming (sliding window) fashion of the input arrival.
   ii. It also offers determinism (if peeks are not chosen) which is needed to guarantee a minimum processing throughput. Sequential control would have difficulties to offer this.
   iii. It limits expressiveness the compute model of my choice would not be Turing complete (as opposed to most sequential control models), but for the problem at hand it is not needed.

Question 3 Reconfigurable System Architectures: (15 points)

a) What is the difference between computer (i) architecture, (ii) implementation, and (iii) realization. Give an example for each of the three levels of abstraction.
b) Which level of abstraction do FPGAs belong to?

Answer

See lecture Reconfigurable System Architectures, slides 4-6.

Question 4 Software for Spatial computing: (15 points)

Consider the following code:

```
if(a>10) {
        a++;}
else{
        a=a*a*a;}
x=a;
```

There are multiple ways to implement this code in spatial computing. Which one would you choose in this specific case and why? Show the hardware block diagram of your implementation.

Answer

Similar to lecture on HLS programming and HW/SW co-design, slides 6-7.

One would choose the true flow control in this case because the two alternative paths have different complexity (thus different expected delays, i.e. an addition is faster than two multiplications).

Question 5 Reconfiguration Management: (10 points)

a) What is the main advantage of partial reconfiguration versus full reconfiguration in modern FPGA devices?
b) Typically, the bitstream of a partial configuration (module) maps to a specific physical location of the FPGA device. How does this limit a reconfigurable system with dynamic partial reconfiguration

Answer

a) Faster, more flexible, may allow reconfiguration to be performed dynamically on the fly
b) It means that relocation and defragmentation options are limited to the number of bitstream versions available for each module.

Question 6 Energy Efficiency: (15 points)

Describe two reasons why spatial computing in Reconfigurable hardware can be more energy efficient than temporal computing in a microprocessor.

Answer

1. **Reduced data movement**: data move from on stage to the next in a dataflow fashion covering much shorter distances compared to traversing the memory hierarchy of a microprocessor.
2. **Lower operating frequency**: dynamic power is proportional to the operating frequency. Reconfigurable designs use in general slower clock ()a few hundred MHz) compared to microprocessors (a few GHz). That means that the dynamic power of reconfigurable designs may be lower than software running on microprocessors (* DVFS in microprocessor chips may minimize this gap).
3. **Higher degree of parallelism**: Spatial computations in reconfigurable hardware may offer higher degree of (data/task/instruction) parallelism, which may speedup the computations at hand. This may reduce execution time and hence the time the device needs to be active compared to software running on a microprocessor, which results in lower energy cost.
4. **Flexibility to adapt to the computations:** A microprocessor needs to be generic and so requires sophisticated mechanisms for efficient control flow and data access. On the contrary, a reconfigurable device can be customized/tailored to the computations at hand. This allows most of the hardware resources to be focused on useful computations (while a

microprocessor spends most of its resources for caches, and complex control, i.e. branch prediction) which can yield better energy efficiency.

Question 7 Defect and Fault Tolerance: (15 points)
Describe the difference between Defects/Permanent Faults and Transient faults (soft-errors). For each of the two types of faults: (i) are FPGAs more vulnerable compared to other chips (ASICs, microprocessors), and (ii) can they tolerate these faults easier than other chips?

Answer
Defects/Permanent Faults: permanent malfunctions due to manufacturing or due to aging, which constitute part of the circuitry on a chip defective.

Transient faults: Faults that happen randomly (due to alpha particles, crosstalk on wires, etc.) at different parts of the chip altering the state of the system.

FPGAs can tolerate better permanent faults because they can be reconfigured.

FPGAs may be more vulnerable to permanent faults if more emerging CMOS technology is used for their implementation (which is common, because of their regular structure). Otherwise, if the same CMOS technology is used, they are equally vulnerable to other chips.

Similar mechanisms used for other chips can be implemented in FPGAs to tolerate transient faults, so it is equally hard to tolerant these faults (if not harder because they are more vulnerable – see bellow).

They are more vulnerable to transient faults because all their configuration bits are stored in some form of memory (i.e. SRAM cells) which can be vulnerable to bit-flips.