# Test 2: Memo

EEE4084F

2016-05-10

Instructions:

- Answer on a separate page.
- Make sure that your student number is on all your answer pages.
- There are 4 questions, each divided into sub-questions. Answer all questions.
- Total time: 35 minutes.
- Total marks: 35.

**Question 1: CPU Architectures** **[8 Total]**

(a) What are some of the advantages of distributed memory over shared memory? Name at least two benefits. **[2]**

**Answer:** Distributed memory systems are more scalable. They are less dependent on a single architecture, for example a cluster of PCs that are not all of the same architecture is possible where a new PC added doesn't necessarily have to have the same exact specifications as the other machines in the cluster.

(b) Briefly explain what is meant by instruction level parallelism (ILP). **[2]**

**Answer:** Instruction level parallelism (ILP) is a measure of how many of the operations in a computer program can be performed simultaneously. The action of executing logically sequential operations at the same time (i.e. in parallel) is called instruction level parallelism.

(c) What is meant by the concept of super-pipelining in regards to ILP? Indicate two potential challenges of designing a processor architecture to support super-pipelining. **[4]**

**Answer:** Superpipelining is an increase in the depth of the pipeline that can shorten clock cycles, giving more instructions in flight at the same time. So on average more than one instruction may be completing per clock cycle. Challenges include:

- For higher degrees of superpipelining, the more forwarding/hazard detection hardware is required, thereby increasing the pipeline latching overhead.
- This pipeline latching overhead can become a big part of the clock cycle time. Potentially greater clock skews may result (as the clocks get faster and faster).

- Superpipelined processors may have longer instruction latency (i.e., cycles/instruction) than simpler processors, which can degrade performance when there are data dependencies. i.e. this approach essentially improves throughput but at a potential expense of latency.
- Superscalar are more susceptible to resource conflicts needing complicated hardware workarounds to make them run correctly.

**Question 2: Reconfigurable Computing** **[10 Total]**

(a) Briefly define what is meant by a digital accelerator (i.e. the type of thing we are trying to prototype in the YODA project). **[2]**

**Answer:** A Digital Application Accelerator is an add-on card / co-processor that supplements a more general purpose processor in order to speed up processing for a particular solution or rather aspects of a computer-based solution. There is an emphasis on digital as such systems support digital processing, the reconfigurable FPGA-based computing solutions typically comprises digital logic/circuitry to speed-up computation. A GPU can be considered an example of a digital application accelerator (focusing on the digital processing aspects of one anyway; although these cards may have additional non-digital components which would not be considered part of the digital accelerator).

(b) Discuss at least two advantages and two disadvantages of using an FPGA-based reconfigurable computing approach to implement a digital accelerator, instead of a more standardised multi-processor CPU-based approach (eg. using a cluster of Intel PC's running MPI)? **[5]**

**Answer:** FPGAs can provide a significant amount of flexibility, supporting a wide range of specialised processing needs, changing the processing structure and data flow to better suite the problem than is allowable by a more standard computer architecture (e.g. Von Neumann) model. It also can support a fine grained solution, where one result is strongly dependent on many data elements at once. An FPGA approach supports a greater level of customizability to change the processing elements to match the chosen solution.

Disadvantages include slower clock speeds, complex compilers, minimal supporting resources, slower data access to the memory, etc.

(c) Briefly explain the difference between a PLA, a CPLD and an FPGA. Don't go into too much detail regarding the internal circuitry – a brief overview of the main differences, in three sentences, is sufficient. **[3]**

**Answer:** These range in complexity from simple to complex; and cheap to expensive. PLA (Programmable Logic Array) is the most simple of the three and therefore least expensive.

The CPLA (Complex PLA) is more complex than the PLA, typically being an arrangement of interconnected PLAs, and is also a bit more costly.

The FPGA (Field Programmable Gate Array) has a more complex arrangement of programmable logic blocks and interconnects, but that comes at additional expense. It is generally the most expensive of these options, but provides support for the largest circuits.

(d) **Bonus mark:** Which company manufactures the small-package and low-power IGLOO FPGA. **[1]**

**Answer:** The IGLOO is a Microsemi FPGA, formally Actel FPGA (but Actel was bought). I would accept either answer for the bonus mark.

**Question 3: Cell Processors** **[4 Total]**

(a) What is the difference between the PPE and the SPE in a cell processor? **[1]**

**Answer:** The PPE (Power Processor Element) is the coordinating master processor in this heterogeneous processor architecture in the system. The SPE (Synergistic Processor Element) provide worker threads connected in various ways together using the Element Interconnection Bus.

(b) Are there more of the one than the other? **[1]**

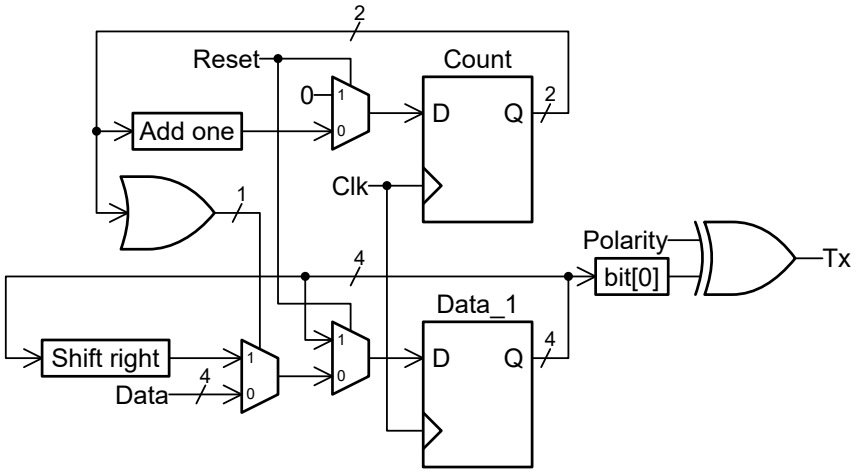
**Answer:** Yes. There is 1 x PPE and 8 x SPEs.

(c) Would you say the interconnection bus is statically configured in connecting the SPE's, or more controllable? Briefly elaborate. **[2]**

**Answer:** In reality it is a combination of both, but each SPE has fancy DMA controllers that coordinate data transfers. The bus structure (at least in earlier versions) comprised a ring network structure whereby neighbouring SPEs in sequence could receive data at high speed and very low latency. Overall, with the complex DMA structures and optimizations, the system is more dynamic than static. However, depending on the elaboration given, I would be willing to mark either static or dynamic correct if it is logically motivated.

**Question 4: Verilog** **[13 Total]**

(a) One possible circuit is: **[8]**




(b) The timing diagram should show at least that the `Polarity` function is asynchronous, the data is registered on the point where `Count` goes from 0 to 1 and data is transmitted least-significant bit first. In the diagram below, blue means "don't care" and red means "unknown". **[5]**