

Assignment 3 : Computer Programming

Date Due: Tuesday, December 5th in tutorial

Question 1: VHDL (10 marks)

Write a VHDL program for a circuit that takes in two inputs: a clock signal called CLOCK and a data signal called DATA. This circuit also has a single output called ALARM.

Design the circuit so that ALARM goes high whenever the last three clocked inputs to DATA have been 101. This is the exact same example that was done in class, so feel free to use the FSM from that class as a reference. For full marks, simulate your VHDL code in Xilinx, and hand in the resulting schematic, along with your VHDL code.

Question 2: Assembler (10 marks)

- a) Write assembler code that calculates the n^{th} Fibonacci number, and prints it to the screen. Assume that this code is a subroutine that begins at location \$1000 in memory, and that the value for n was stored in data register d0 before the subroutine was called.
- b) Write assembler code that performs the middle-of-three pivot calculation for Quicksort. Registers a0, a1 and a2 contain the addresses for three long signed integers in memory. Calculate the middle of these three values, and write the result into the location in memory specified by a4 before returning from the subroutine.

Question 3: Microcode (10 marks)

The following assembly language instructions redirect the control of the program, depending on the contents of A0:

```
foo      SUB.L    (A0),D0
          BLT     foo
          JMP     $12345
```

Using Appendix C of the text (or some equivalent instruction code tables), write the opcodes for these instructions.

Question 4: Datapath (10 marks)

Consider the assembly language instructions from Question 3. If the second line of `foo` does not branch back to the first (i.e. the contents of `A0`'s memory location was less than the value within `D0`), give the datapath signals that will take place when carrying out the operation. Assume the single-bus architecture that was used in class (provided below), and reduce the number of steps needed as much as possible. Start by fetching the first instruction (`SUB.L`).

