

ELEC-E3540 Digital Microelectronics II – Exercise 6

Book: Peter J. Ashenden, “The designer’s guide to VHDL”, 3rd edition

Acquaint yourself with the following chapters: 3.1-3.5, 2.2.5, 5.2, 21.5.

Things to learn:

- State machines
- If and case
- Assert

Pre-exercise tasks: Get acquainted with the PIC16F84A datasheet. Your task is to design the instruction decoder as a VHDL state machine. The decoder should support only the instructions that are to be implemented for the design assignment (this was defined during the intro lecture).

In PIC16F84A the execution of every instruction can be divided into “states”. Maximum number is four, since PIC datasheet describes execution in max four clock cycles. Only memory write is strictly synchronous operation, but in order to make things easier, advice is to implement the steps with a synchronous state machine. Take into account that every command does not require every step. You may control this by defining your next states according to operation.

The states of the operation execution are:

1. **iFetch:** Read instruction from the program memory and decode it. Decoding means determining the opcode and the source of input arguments (i.e. literal, register or memory).
2. **Mread:** Read operand from memory, if required.
3. **Execute:** Perform operation with ALU.
4. **Mwrite:** Write data to memory/register, update status flags, increment program counter.

For simplicity, you can assume that the program counter is always incremented, and therefore cannot be the destination of a byte-oriented instruction (this would be equivalent to a branch).

Sketch on paper the “skeleton” of the instruction decoder state machine. For each operation, determine the required states, the source of the operands, and the targets of **Mwrite**.

Write the VHDL of the synchronous state machine that takes care of the operation decoding. Use the “assert” construct to indicate cases where normal writing of status register is overridden.

I strongly suggest that you do most of the coding before the actual exercise time!

Exercise task: Complete the synchronous VHDL state machine that takes care of the instruction decoding and execution. Write a suitable test bench to test the operation. Simulate.

Goal: Student has learned basics of synchronous and combinatorial logic, and the design of finite state machines with VHDL.

Workload: Preparations 8h + exercise 2h