ELEC-E3540 Digital Microelectronics II – Exercise 4

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

Acquaint yourself with the following chapters: 6.1-6.6.

Things to learn:

- Procedures and functions
- Structure of ALU of the PIC16F84A microcontroller

Pre-exercise task: Design an ALU (i.e. sketch, draw, and write as much code as you can) that is compliant with the PIC16F84A microcontroller. The structure of the ALU is as follows:

- ALU has two possible 8-bit input operands.
- The operation is selected with signal *operation*, which is declared from a custom enumeration type.
- The result of the operation is the output *result* of the ALU.
- The position of the bit-oriented operations is indicated with 3-bit signal *bit_select*.
- The ALU is asynchronous (i.e. purely combinational circuit, any memory elements are external).
- Output *status* equals the next state of the status register.
- Input *status_in* equals the current state of the status register.
- Status register consists of 3 bits, described as follows
  - bit 2 Z: Zero bit
    ’1’ = The result of an arithmetic or logic operation is zero
    ’0’ = The result of an arithmetic or logic operation is not zero
  - bit 1 DC: Digit carry/borrow bit
    ’1’ = A carry-out from the 4th low order bit of the result occurred
    ’0’ = No carry-out from the 4th low order bit of the result
  - bit 0 C: Carry/borrow bit
    ’1’ = A carry-out from the most significant bit of the result occurred
    ’0’ = No carry-out from the most significant bit of the result occurred
    Note: For rotate (RRF, RLF) instructions, this bit is loaded with either the high or low order bit of the source register.

The ALU should be capable of executing all the arithmetic and logic operations described by PIC16F84A datasheet. Rest of the instructions are handled by the ALU as NOP. Also, beware that the ALU does not make distinction between, e.g., ADDWF and ADDLW (that will be handled by the instruction decoder).

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

Exercise task: Write the ALU description and a test bench for it. Use functions and procedures instead of cut-and-paste in places where the same operation is performed several times.

Read the values of input signals from a “program file” that consists of some coded operations and required operands for ALU. One operation per row.

Write a test bench that reads the “program file”, executes it with ALU, and writes all the outputs to another file.

Goal: Student has learned basics about procedures and functions, and applied his/her VHDL skills to a typical microcontroller ALU.

Workload: Preparations 8h + exercise 2h