



MODULAR PROGRAMME  
ASSESSED COURSE-WORK SPECIFICATION

**Module Details:**

Module Code: <b>UFMEVP-20-2</b>	Module Title: <b>Architecture of CPUs with VHDL</b>	
Module Leader: <b>Nigel Gunton</b>		
Module Tutors: <b>Nigel Gunton</b>		
Assignment <b>CW3</b>	Element Number: Weighting <b>38%</b>	Total Assignment Time: <b>18++ hrs</b>

**Dates:**

Date assignment issued to students: <b>wb Nov 9th '09</b>	Date for return of marked work: <b>wb May 18th 2010</b>
Submission Place: <b>Project Room, 2Q30 open 09.00-18.00</b>	Date of Submission: <b>April 29th 2010</b>
	Time of Submission: <b>14.00</b>

**Deliverables:**

<b>As listed on the Assignment spec sheet</b>
---

## 1. Stage Two :

Completed and verified control unit:

VHDL CODE  
Test Results  
Demonstration

- 10 marks

Stage Three: ALU, Registers & System Integration (individual or pair)

- 1 Develop and test the ALU as discussed in the lectures.
- 2 Develop and test the various registers that are required.
- 3 Integrate the CU, the ALU and the registers into a CPU and test with the program that will be provided. This is a straightforward but time consuming task. You should verify that the integration is functionally correct by writing simple test patterns before running the example program.

### 1.1. Deliverables

- 1) A written test plan describing how you intend to verify the correct working of the ALU, test results for both functional (dataflow) and timing (structural) models, along with the .vhd code.

Test plan for ALU  
Test Results : Functional model  
Test Results : Timing model  
VHDL Source

-  
- 15 marks

- 2) Provide one example of your CPU register designs along with the test plan and results. Your selection must include tri-state outputs.

Test plan for Register  
Test Results : Functional model  
Test Results : Timing model  
VHDL Source

- 10 marks

- 3) Provide the top level (hierarchical) code along with a screenshot from `xsch` of your integrated CPU, the results of your verification tests and the results of the test code simulation. (Even if the simulation failed)

Top level CPU code  
Screenshot  
Your test plan  
Your test plan results  
Supplied test program results

- 20 marks

## 2. Stage Four: Further development (individual or pair)

Select **ONE** of the following options :

- a) Extend the instruction set and the register set to support a stack pointer, additional status flags, eg `carry` and `overflow`, along with call and return instructions. Write a simple program to test your extended instruction set. Your modified processor should also pass the test from phase 2:part 4.

or

- b) Port the design to an FPGA using the Altera Quartus software. You should develop a complete system on chip, using the Quartus library to provide ROM, RAM, glue logic and I/O.

Other extensions may be considered. Discuss this with your lab tutor before proceeding.

### 2.1. Deliverables

- 1) The allocation of marks will be in-line with the other deliverables. The exact nature of deliverables for Stage 4 can be discussed with your lab tutor. The marks for Stage 4 will be out of 30. A typical distribution is

Design documents	- 10 marks
Test Plan	- 5 marks
Implementation	- 10 marks
Test results	- 5 marks

The weighting between design and implementation will depend on whether option (a) or option (b) is chosen.

### 3. Document Quality

The overall quality of the final hand in will be considered when marking.

Document Quality	- 15 marks
------------------	------------