



MODULAR PROGRAMME
ASSESSED COURSE-WORK SPECIFICATION

Module Details:

Module Code: UFMEVP-20-2	Module Title: Architecture of CPUs with VHDL	
Module Leader: Nigel Gunton		
Module Tutors: Nigel Gunton		
Assignment CW1	Element Number: Weighting 19%	Total Assignment Time: 9 hrs

Dates:

Date assignment issued to students: wb Oct 17th '11	Date for return of marked work: 6th Jan '12
Submission Place: Project Room, 2Q30 open 09.00-18.00	Date of Submission: 24th Nov 11
	Time of Submission: 14.00am

Deliverables:

As listed on the Assignment spec sheet

1. Phase One: Addressing Modes & Architectures (Individual Component).

Each student must provide an individual report for this phase of the assignment. You must use the Harvard referencing style for all your sources. Direct 'copy&paste' from Wikipedia or similar into the body of your report, other than for literal quotes where appropriate, will accrue ZERO marks.

2. Specification:

A small microprocessor is to execute the task shown on the next page. The programme consists of an initialisation phase followed by a continuous cycle of

- reading from an input port,
- using the data received to index into an array of constant values
- and then to write the data retrieved from the array to an output port.

The initialisation phase copies the main programme code from ROM into RAM and then jumps to the start point in RAM.

- a) What addressing modes would be required in an instruction set if it were to be used to implement this program.
- b) Select a processor from the list below and provide a short description of the register set from a programmers perspective along with a summary of the addressing modes provided.
- c) With respect to your chosen processor assess to what extent it supports the addressing modes required in question a. How might you overcome any shortcomings if it does not support the required addressing modes?

2.1. Deliverables

An individual report, not exceeding 1500 words, that answers questions a, b &c above. Marks will be awarded for quality of presentation, clarity of your descriptions and analysis of the suitability of the selected processor. It is acceptable if your chosen processor is not suitable for the task required provided that your justification is well reasoned. You should also consider the memory map outlined in appendix 2 and whether the processor under discussion can accommodate it.

3. Processor List

It is recommended that you select a processor from the following list. Note that although some of the processors listed have been in existence for some considerable time, they are still in active production. They may also be available as synthesisable IP cores. You must justify your reasons if you choose to use a different processor. You should select a processor that uses a Von Neumann architecture http://en.wikipedia.org/wiki/Von_Neumann_architecture

- 6502 <http://www.6502.org/>
- ZPU <http://opencores.org/project,zpu>
- Rabbit 2000 http://en.wikipedia.org/wiki/Rabbit_2000
- Z80 http://en.wikipedia.org/wiki/Zilog_Z80

4. Appendix 2

Program specification & memory map (8 bit data, 16 bit Address)

- The program MUST reside in ROM and WILL be relocated to RAM after booting.
- The program is to read an 8 bit wide value in from an input port.
- The value is used as an index into an array held in ROM.
- The data (also 8 bits wide) read from the array is to be written to an output port.

An example memory map:

```
0x8080 DATA_IN
0x8000 DATA_OUT
0x07FF RAM_END
0x0400 RAM_START
0x03FF ROM_END
0x0000 ROM_START
```

Programme algorithm:

```
BEGIN
  FOR i = 0; i < length of programme; i++ DO
    COPY PROGRAMME CODE FROM ROM TO RAM
  DONE;

  FOR ever DO
    LOAD DATA_IN, register;
    STORE array[DATA_IN], DATA_OUT;
  DONE;
END;
```

