

<b>Module Title:</b>	Digital Systems
<b>Language of Instruction:</b>	English

<b>Credits:</b>	10
-----------------	----

<b>NFQ Level:</b>	8
-------------------	---

<b>Module Delivered In</b>	<a href="#">2 programme(s)</a>
----------------------------	--------------------------------

<b>Teaching &amp; Learning Strategies:</b>	A combination of lectures, class discussion, tutorial, laboratory exercises and demonstrations will be used. Emphasis will be placed on active learning including problem / project bases learning.
--	---

<b>Module Aim:</b>	To provide learners with the knowledge and skills needed to design, simulate, synthesise and validate digital systems using SOC technology.
--------------------	---

### Learning Outcomes

*On successful completion of this module the learner should be able to:*

LO1	Describe System on Chip (SoC) architectures and interfaces.
LO2	Use contemporary Electronic Design Automation (EDA) tools to develop digital systems using field programmable SOC devices.
LO3	Use a HDL to create a self-checking test bench to test a digital system which incorporates combinational and sequential logic.
LO4	To use a SOC bus technology to interconnect system components.
LO5	Design and implement a small system incorporating a datapath and datapath controller.
LO6	Analyze systems specifications in order to plan, partition, design, implement and test a digital system.

### Pre-requisite learning

#### **Module Recommendations**

*This is prior learning (or a practical skill) that is recommended before enrolment in this module.*

No recommendations listed

#### **Incompatible Modules**

*These are modules which have learning outcomes that are too similar to the learning outcomes of this module.*

No incompatible modules listed

#### **Co-requisite Modules**

No Co-requisite modules listed

#### **Requirements**

*This is prior learning (or a practical skill) that is mandatory before enrolment in this module is allowed.*

Hardware Description Language or equivalent

**Module Content & Assessment**

**Indicative Content**

**Datapaths:**

Use a HDL to synthesize/simulate datapath elements (adders, multipliers, shift registers, ALU etc.), zero/sign extension, carry/overflow, casting (sign, size). Incorporating a pipeline (latency, throughput). Datapath controller – review of state machines. Development of a subsystem consisting of a datapath & datapath controller (eg. Shift and add multiplier, RPN calculator etc.)

**Memory:**

FPGA memory resources (embedded memory & distributed logic cells). Use a HDL to synthesis/simulate RAM based components (single/dual port synchronous RAM, register file, Stack, FIFO). Use a HDL to synthesize/simulate systems that incorporate ROM (lookup-table, linearization, waveform generator). Memory initialization file.

**Clock:**

Clock generation (PLL), management and distribution. Clock trees, clock skew, clock regions, power optimization. Clock enable & clock\_division (used for synchronous clocking at very low frequency).

**System design:**

Getting from a specification to system. Partitioning the system - block diagram, dataflow diagram, timing diagrams. Hardware/software co-design.

**CPU:**

Overview of options for CPU - Softcore: (Microblaze, Nios, Cortex-M1, Risc-V). Hardcore (Cortex A series, Risc-V). Developing code for the embedded CPU of a selected SOC. Operating system (PetaLinux or UcLinux?).

**Buses:**

Overview of SOC bus technologies (AMBA, AXI, CoreConnect, Wishbone). Use a chosen bus to interconnect system components (IP). Handshake, decoding, arbitration, clocking.

**Testbench:**

Creating a self checking test bench. Layered testbench, scoreboarding, assertions, classes, constrained random variables.

**Systems issues:**

Project - Development of a system that utilizes a CPU, Buses, IP and OS. Investigate the implications of design approach/partitioning on power, speed, timing and area. Static timing analysis.

**Misc:**

Automation of the build process using a script language such as Tcl/Tk.

Assessment Breakdown	%
Continuous Assessment	20.00%
Project	30.00%
End of Module Formal Examination	50.00%

**Continuous Assessment**

Assessment Type	Assessment Description	Outcome addressed	% of total	Assessment Date
Examination	n/a	1,2,3	20.00	n/a

**Project**

Assessment Type	Assessment Description	Outcome addressed	% of total	Assessment Date
Project	n/a	2,3,4,5,6	30.00	n/a

No Practical

**End of Module Formal Examination**

Assessment Type	Assessment Description	Outcome addressed	% of total	Assessment Date
Formal Exam	n/a	1,2,3,5	50.00	End-of-Semester

**Module Workload**

<b>Workload: Full Time</b>		
<i>Workload Type</i>	<i>Frequency</i>	<i>Average Weekly Learner Workload</i>
Lecture	Every Week	4.00
Laboratories	Every Week	3.00
Independent Learning	Every Week	5.00
Total Hours		12.00

**Module Delivered In**

Programme Code	Programme	Semester	Delivery
CW_EEBEE_B	<a href="#">Bachelor of Engineering (Honours) in Biomedical Electronics</a>	8	Mandatory
CW_EESYS_B	<a href="#">Bachelor of Engineering (Honours) in Electronic Engineering</a>	8	Mandatory