Next Generation Higher National Unit Specification

Digital Electronics: Theory and Applications (SCQF level 7)

Unit code: J752 47

SCQF level: 7 (24 SCQF credit points)

Valid from: session 2023–24

Prototype unit specification for use in pilot delivery only (version 1.0) April 2023

This unit specification provides detailed information about the unit to ensure consistent and transparent assessment year on year.

This unit specification is for teachers and lecturers and contains all the mandatory information required to deliver and assess the unit.

This edition: April 2023 (version 1.0)

Unit purpose

This unit provides learners with knowledge and skills specific to electronics engineering.

They learn about the principles underlying digital electronics, including:

- combinational logic
- sequential logic
- linear computer programs
- non-linear computer programs

The target learner group for this unit is learners who want to develop their core electronic engineering skills to support a career in electronic engineering

Entry to the unit is at your centre's discretion. However, we recommend that learners have one or more of the following:

- broad knowledge and understanding of electronics concepts and theorems; for example, an SCQF level 6 qualification in Electronics Principles, Higher Physics or a National Certificate in an engineering discipline
- relevant, equivalent workplace experience

The unit provides learners with suitable knowledge and skills to progress to further study, or employment in a wide range of engineering industries.

Unit outcomes

Learners who complete this unit can:

1 apply knowledge and skills of digital logic design and tools to combinational logic circuits
2 apply knowledge and skills of digital logic design and tools to sequential logic circuits
3 apply knowledge and skills of high-level language and numbering systems to firmware linear coding applications
4 apply knowledge and skills of high-level language and numbering systems to firmware non-linear coding applications
5 describe the applications of digital electronics

Evidence requirements

You can assess this unit holistically using a portfolio of evidence that learners generate. Learners must produce a reflective report for each outcome, evaluating the knowledge and skills they have gained.

Evidence should principally consist of written and/or oral recorded evidence. Learners generate evidence under unsupervised, open-book conditions.

You can find further information in the 'Additional guidance' section.

Outcome 1

Combinational logic

1a Using a truth table and Karnaugh map to minimise complexity, and a circuit diagram, design a combinational logic circuit that uses at least three different types of gates from AND, OR, NOT, NAND, NOR and EXOR.

1b Simulate and evaluate the circuit designed in 1a. Learners must demonstrate that the circuit functions correctly.

1c Using Boolean algebra to minimise complexity and a circuit diagram, redesign the circuit in 1a to use only two-input NAND gates.

1d Simulate and evaluate the circuit designed in 1c. Learners must demonstrate that the circuit functions correctly.

1e Build, test and evaluate a hardware combinational logic circuit that uses devices from both transistor to transistor logic (TTL) and complementary metal oxide semiconductor (CMOS) families. The complexity of the circuit should be equivalent to 1a and may be the same circuit. Learners must demonstrate that the circuit functions correctly.

1f Document 1a to 1e, including evidence of innovation, self-management, interfacing between different logic families, and the use of technical datasheets.

Outcome 2

Sequential logic

2a Using a state transition diagram, action table, excitation table and Karnaugh map to minimise complexity, and a circuit diagram, design an asynchronous sequential logic circuit that uses at least three D-type bistables and combinational logic.

2b Simulate and evaluate the circuit designed in 2a. Learners must demonstrate that the circuit functions correctly.

2c Using a state transition diagram, action table, excitation table and Karnaugh map to minimise complexity, and a circuit diagram, design a synchronous sequential logic circuit that uses at least three JK flip-flops and combinational logic.

2d Build, test and evaluate a hardware synchronous sequential logic circuit that uses at least three JK flip-flops and combinational logic. The circuit may be the same as 2c. Learners must demonstrate that the circuit functions correctly.

2e Document 2a to 2e, ensuring that documentation includes evidence of innovation, self-management and the use of technical datasheets.

Outcome 3

Linear program

3a Use a recognised software design process, such as pseudocode or a flow chart, to convert a software specification to a detailed linear design.

3b Using the design process in 3a, write, simulate and evaluate one linear program containing:

- a code library
- code comments
- version control evidence
- two different arithmetic operators (for example, +, −, *, /)
- one external input that requires configuration and reading from
- one external output that requires configuration and writing to
- two different data types (for example, integer, character, long integer, float, string)
- two different number systems (for example, binary, hexadecimal, octal)
- one variable
- one constant
- one bit-wise operation (for example, &, |)
- output to one user interface display (for example, a serial monitor)

Learners must demonstrate that the code functions correctly.

3c Write a test specification that comprehensively tests the program written in 3b. The test specification should evidence an understanding of hardware constraints (for example, interfacing, power, speed, space, memory allocation).

3d Upload, test and evaluate the program written in 3b to the test specification written in 3c. Learners must demonstrate that the hardware functions correctly.

3e Document 3a to 3d, ensuring that documentation includes evidence of innovation, self-management and an understanding of number system conversion.

Outcome 4

Non-linear programs

4a Use a recognised software design process, such as pseudocode or a flow chart, to convert a software specification to a detailed non-linear design.

4b Using the design process in 4a, write, simulate and evaluate one non-linear program containing:

- a code library
- code comments
- version control evidence
- two different relational operators (for example, ==, !=, >, <, >=, <=)
- one compound Boolean expression that uses two or more different logical operators (for example, AND, OR, EXOR)
- one branch statement, with two or more branches (for example, if-then-else)
- one multi-way selection statement with three or more branches (for example, switch-case)

Learners must demonstrate that the code functions correctly.

4c Use a recognised software design process, such as pseudocode or a flow chart, to convert a software specification to a detailed non-linear design.

4d Using the design process in 4c, write, simulate and evaluate one non-linear program containing:

- a code library
- code comments
- version control evidence
- one iterative process (for example, while, until, for-do)
- one iterative loop that manipulates an array (for example read port data into an array, send array data to a port, read from an array, write to an array)
- one interrupt service routine (ISR)
- one user-defined function

4e Document 4a to 4d, ensuring that documentation includes evidence of innovation, self-management and an understanding of how the processor uses the code to make decisions.

Outcome 5

Emerging technologies

Describe how digital electronics is used in, and influences, two technologies, as relevant to contemporary developments.

Knowledge and skills

The following table shows the knowledge and skills covered by the unit outcomes:

Knowledge	Skills
Learners should understand how to: ♦ design combinational logic circuits using truth tables, Karnaugh maps, Boolean algebra and circuit diagrams ♦ design sequential logic circuits using state transition diagrams, Karnaugh maps, action tables and circuit diagrams ♦ write and simulate linear programs using a high-level language, and knowledge of binary, octal and hexadecimal numbering ♦ write and simulate non-linear programs using a high-level language, and knowledge of binary, octal and hexadecimal numbering ♦ describe the role that digital electronics plays in emerging technologies, for example in artificial intelligence, virtual reality, cloud computing, computer vision and other contemporary technologies	Learners can: ♦ build, test and evaluate combinational logic circuits, in hardware, using circuit diagrams and technical datasheets for different logic families ♦ build, test and evaluate sequential logic circuits, in hardware, using circuit diagrams and technical datasheets for different logic families ♦ upload, test and evaluate linear programs, in hardware ♦ upload, test and evaluate non-linear programs, in hardware

Meta-skills

Throughout this unit, learners develop meta-skills to enhance their employability in the engineering sector.

Self-management

Learners develop the meta-skills of focusing, adapting and initiative, as they test and modify circuits and software. They can provide evidence of these in their technical reports.

Social intelligence

Learners develop the meta-skills of communicating, collaborating and leading, as they collaboratively develop circuits and code. They can provide evidence of this by working with others in supervised practical activities.

Innovation

Learners develop the meta-skills of curiosity, creativity, sense-making and critical thinking as they design circuits and software. They can provide evidence of these by explaining their designs in their technical reports.

Literacies

Learners develop core skills in the following literacies:

Numeracy

Learners develop their numeracy by using binary, octal and hexadecimal numbers in the code they write.

Communication

Learners develop their communication skills by presenting test results and working with peers to develop circuits and software.

Digital

Learners develop digital literacy by using a broad range of engineering software and testing equipment.

Delivery of unit

This unit is part of HNC Engineering. The framework includes mandatory and optional units, and you can tailor the selected combination of units to specific engineering pathway needs. You can deliver this unit alongside, and use it to support, the SCQF level 7 unit, Electronics in Practice: Skills and Techniques.

While the exact time allocated to this unit is at your centre's discretion, the notional design length is 120 hours.

The amount of time you allocate to each outcome is also at your discretion. We suggest you spend 30 hours on each outcome, including assessment.

You should give learners formative opportunities to design, simulate, build and test circuits and programs that are not for summative, final assessment.

Additional guidance

The guidance in this section is not mandatory.

Content and context for this unit

This unit gives learners some of the knowledge and skills they need to support a career in electronics engineering.

Apply knowledge and skills of digital logic design and tools to combinational logic circuits (outcome 1)

Learners develop the ability to design, optimise, simulate, test and evaluate combinational logic circuits, including those used to control simple processes such as conveyor belt control. Learners develop the ability to use the following tools required to do this: truth tables, Boolean algebra, Karnaugh maps, circuit diagrams and simulators. They compare virtual designs with real-world hardware, and learn the constraints imposed by different logic families, the interfacing required between these, and how to use technical datasheets. In testing and documenting their designs, learners develop their innovation, communicating and self-management skills.

Apply knowledge and skills of digital logic design and tools to sequential logic circuits (outcome 2)

Learners develop the ability to design, optimise, simulate, test and evaluate sequential logic circuits, such as those used to control more complex processes, for example analogue to digital conversion. Learners develop the ability to use the tools required for this: state transition diagrams, action tables, excitation tables, Karnaugh maps and circuit diagrams. They learn the differences between D-type and JK flip-flops, and between asynchronous and synchronous logic. In testing and documenting their designs, learners develop innovation, communication and self-management skills.

Apply knowledge and skills of high-level language and numbering systems to firmware linear coding applications (outcome 3)

Learners develop the ability to write, simulate, test and evaluate simple linear programs, in a high-level language. They use recognised good practice in software design and documentation, including the use of flow charts or pseudocode, version control, code commenting, structured testing and user display. Learners develop appreciation of the difference between simulators and hardware, by consideration of hardware constraints. They develop practical testing skills, and mathematical skills in the numbering systems used in digital electronics.

Apply knowledge and skills of high-level language and numbering systems to firmware non-linear coding applications (outcome 4)

Learners develop the ability to write, simulate, test and evaluate non-linear programs, in a high-level language, such as those used in simple robotics. They learn how conditional code, interrupt service routines and iterative processes allow machines to make decisions. They use recognised good practice in software design and documentation. They also develop innovation, communication, self-management and practical testing skills.

Describe the applications of digital electronics (outcome 5)

Learners develop an appreciation of the importance of digital electronics in real-world emerging technologies. They can contextualise their learning in outcomes 1 to 4.

Required resources

The resources that each learner requires for outcomes 1 and 2 are:

- digital logic simulation software
- breadboards
- single strand insulated jumper wires
- wire preparation tools
- basic combinational logic devices families (AND, OR, NAND, NOR, NOT, EXOR) from TTL and CMOS logic families
- sequential logic JK flip-flops
- digital logic test equipment (for example, logic probes)

The resources that each learner requires for outcomes 3 and 4 are:

- an integrated development environment (IDE)
- a compatible processor with upload capability and user-configurable ports
- a means of generating inputs and interrupts (for example, switches), and reading outputs (for example, light-emitting diodes)
- a means of reading code-generated messages (for example, a serial monitor)

The hardware resources for all outcomes may be available as kits. Learners studying the unit remotely need to purchase or hire suitable hardware, power sources and test equipment.

Approaches to delivery

You should deliver outcomes 1 to 4 in order, as they progressively build up knowledge and skills. However, you can deliver outcome 5 at any time.

You should deliver the unit in a physical learning space or virtual learning environment (VLE). Using the holistic teaching format of active learning encourages learners to consider the deeper context of the theory.

This unit is best suited for on-campus delivery and would benefit from technician support to set up equipment, and resolve equipment and hardware failure. However, you can deliver it remotely if learners are well supported with software, hardware, health and safety requirements, and online synchronous lecturer time. You should take care to ensure equitable experience. Cloud computing applications can benefit simulation, as they allow lecturers, learners and peers to share designs with each other.

Approaches to assessment

In line with the approach to delivery, you should take a holistic approach to assessment. For example, learners can generate evidence for outcomes 1 and 2 through one complex design, and for outcomes 3 and 4 through another. You can assess outcome 5 by using a consistent theme throughout. Alternatively, you could use a single project with staged milestone submissions to assess all five outcomes.

You should use formative assessment of practical and written work in all outcomes to develop a learning cycle of assessment and feedback.

You can assess outcomes individually; however, if you take a project-based delivery approach, assessments are likely to be more authentic and closer to real-life applications.

You should encourage learners to work with peers on designs and testing, without compromising individuality. You can assess their ability to collaborate in a group by their contribution to, and reflection on, group work. If designs are shared, for example, using cloud computing applications, you should stress to learners the importance of avoiding plagiarism.

Checklists or rubrics may be useful to ensure learners have achieved all knowledge and skills required by the learning outcomes, as well as meta-skills and literacies.

Learners should collate all evidence in their individual portfolio. This should include reflective accounts to measure their meta-skills, digital literacies, professional skills and wider employer-desired skills. They could record this in their personal portfolio. In particular, you can use reflective reports to assess learners' integrity. You should provide learners with support, guidance and feedback on areas of development, and signpost developmental opportunities.

Because of the open-book nature of the assessment, you must take care to ensure authenticity. You could do this by using variable values in coursework, making use of oral questioning and using originality-checking software. Remote learners may upload videos of operating circuits to their portfolios as evidence.

Opportunities for e-assessment

Assessment that is supported by information and communication technology (ICT), such as e-testing or the use of e-portfolios or social software, may be appropriate for some assessments in this unit.

If you want to use e-assessment, you must ensure that you apply the national standard to all evidence and that conditions of assessment (as specified in the evidence requirements) are met, regardless of the mode of gathering evidence.

Equality and inclusion

This unit is designed to be as fair and as accessible as possible with no unnecessary barriers to learning or assessment.

You should take into account the needs of individual learners when planning learning experiences, selecting assessment methods or considering alternative evidence.

Guidance on assessment arrangements for disabled learners and/or those with additional support needs is available on the assessment arrangements web page: www.sqa.org.uk/assessmentarrangements.

Information for learners

Digital Electronics: Theory and Applications (SCQF level 7)

This information explains:

- what the unit is about
- what you should know or be able to do before you start
- what you need to do during the unit
- opportunities for further learning and employment

Unit information

This unit provides you with knowledge and skills specific to electronics engineering. It is part of HNC Engineering, which is aimed at learners who wish to become engineering technicians.

Unit outcomes

On completing the unit, you are able to:

1. apply knowledge and skills of digital logic design and tools to combinational logic circuits
2. apply knowledge and skills of digital logic design and tools to sequential logic circuits
3. apply knowledge and skills of high-level language and numbering systems to firmware linear coding applications
4. apply knowledge and skills of high-level language and numbering systems to firmware non-linear coding applications
5. describe the applications of digital electronics

In outcomes 1 and 2 of the unit, you learn how to design combinational and sequential logic circuits to perform engineering tasks, such as controlling a conveyor belt. Using design aids, such as truth and action tables, Karnaugh maps, Boolean algebra, and circuit diagrams, you optimise your designs and check they are working using industry-standard simulation software. With a knowledge of the different logic families, and how they interface with each other, you then build and test real circuits, using information from manufacturers' component datasheets.

In outcomes 3 and 4, you extend your knowledge from outcomes 1 and 2 to write programs for a more complex processor circuit. Using a high-level programming language, you write code to control engineering processes, including computer decision making. You learn the principles behind good software design and how to test code to ensure it meets a specification. Using an industry-standard integrated development environment (IDE) to develop your programs, you go on to upload your code to a real processor and prove its functionality with test equipment as used by digital electronics engineers. You learn the limitations that real hardware presents to a fully simulated and working solution.

In outcome 5, you study a variety of contemporary emerging technologies to learn how digital electronics plays a major role in the development of new and exciting technical advances.

You are assessed holistically, meaning your knowledge and skills of the unit content are assessed in one or more engineering scenarios. A variety of assessment approaches are used to assess both practical work and written work, and this includes completing project work. You should record all evidence for the unit in your individual portfolio.

Meta-skills

Throughout the unit, you develop meta-skills to enhance your employability in the engineering sector.

Meta-skills include self-management, social intelligence and innovation.

Self-management

You develop the meta-skills of focusing, adapting and initiative as you test and modify circuits and software.

Social intelligence

You develop your communicating, collaborating and leading meta-skills as you collaboratively develop circuits and code.

Innovation

You develop the meta-skills of curiosity, creativity, sense-making and critical thinking as you design circuits and software.

Administrative information

Published: April 2023 (version 1.0)

Superclass: XL

History of changes

Version	Description of change	Date

Note: please check SQA's website to ensure you are using the most up-to-date version of this document.