CSSE2010/CSSE7201 — Introduction to Computer Systems

Table of contents

Staff . . . 1
Textbook . . . 1
Study formula . . . 2
Course structure . . . 2
Course learning activities . . . 2
Assessment . . . 3
Resources . . . 5
Lectures . . . 5
Exercises . . . 7
Sample mid-semester exams . . . 9
Reference notes . . . 9

Staff

- **Dr Martin Ploschner** — course coordinator & lecturer
- **Dr Chamith Wijenayake** — course coordinator & lecturer
- Email (both coordinators): csse2010@eecs.uq.edu.au
- Office hours: listed on Blackboard
- A large group of tutors support the Learning Lab sessions

Textbook

No textbook required — some references are listed in the course profile. The **course profile** itself is required reading: it is the authority on assessment and policy.

Study formula

1L-2P-1L-2P-3S-1E per week (L = Lecture, P = Practical, S = Self study, E = Exercise/Problem solving).

Course structure

The course builds up through the levels of computer abstraction (see binary-number-representations[1] and logic-gates[2] for the Level 0 material covered first). It runs in three blocks:

Weeks	Block
W01–W05	Hardware (digital logic, gates, AVR microcontroller hardware)
W06–W08	Low-level software (assembly language)
W09–W13	High-level software

The six numbered levels of abstraction the course works up through are drawn in 2026-07-27-course-intro-bits-bytes-binary[3].

Course learning activities

- In-person mode only.
- **Lectures**: Mon 2pm AEST and Thu 8am AEST.
- **Learning Labs** (practical sessions): in-person at 47-104, two sessions per week.
  - Sign up for a *pair* of lab sessions — [PRA-x-P2, PRA-x-P1] — 7 pairs to choose from.
  - PRA-x-P2 sessions run Mon-Tue; PRA-x-P1 sessions run Thu-Fri.
  - **No P2 sessions in week 1** (Mon-Tue) — week 1 labs start from Thursday (P1 sessions only).
  - From week 2 onward, both P2 and P1 sessions run every week.
  - Sign-on issues: contact eait.mytimetable@uq.edu.au.
- A lab kit is loaned **during weeks 1 to 4**, collected in person during lab classes, and must be returned by the end of the course in working order (or within 3 business days of dropping the course, to the EECS school office at 78-425). **You must have a lab kit**

[1] courses/csse2010/binary-number-representations.pdf
[2] courses/csse2010/logic-gates.pdf
[3] courses/csse2010/2026-07-27-course-intro-bits-bytes-binary.pdf

with you in order to complete and pass the course. See io-board[4] for the board it contains.
- Lectures contain interactive sessions requiring **an electronic device** (laptop/tablet/phone).
- Labs program the kit **preferentially in Microchip Studio**. Alternative tools are provided for Mac users, but **formal assessments are based on Microchip Studio**.
- Practice exercises and weekly quizzes are provided but **not assessed**.

Assessment

Component	Weight	Due	Notes
In-semester exam	20%	In-semester Saturday, 5/09/2026 – 20/09/2026	Personal exam timetable emailed
Lab exam (Digital Logic Design)	10%	Week 7	In-person
AVR Programming Assignment	20% (CSSE2010) / 5% + 15% (CSSE7201)	23/10/2026, 4:00pm	**Hurdle**
AVR Programming Assignment (Demo)	Pass/Fail	Week 13	**Hurdle**
Final Examination	50%	End of semester exam period, 7/11/2026 – 21/11/2026	**Hurdle**

- **CSSE2010**'s AVR Programming Assignment is Practical-only (20%). **CSSE7201**'s splits into a Theoretical Part 1 (5%) and Practical Part 2 (15%). The course profile is CSSE2010-only.
- The four exams and the demo are in person; the in-semester exam, lab exam, demo and final are **identity verified**, the assignment is not.
- **The assignment mark is scaled by your demo performance**, and **you must attend the demo or the assignment scores 0** — the demo is not just a separate pass/fail box.
- The assignment is an **open assessment**: notes, course materials and AI are permitted. Everything else is secure.

The final exam hurdle

Two thresholds, and both bite:

[4] courses/csse2010/io-board.pdf

- **Grade 3** (i.e. to be eligible for supplementary assessment): **35% on the final exam**. Below that, your grade is **capped at 2**.
- **Grade 4** (i.e. to pass): **40% on the final exam AND 50% overall.**

Grade cutoffs

Grade	Cumulative %
7	85
6	75–84
5	65–74
4	50–64
3	45–49
2	20–44
1	$<$ 20 (with at least one item attempted)

The cumulative percentage is the sum out of 100, rounded, then capped. Three capping scenarios exist in the profile (caps at 44, 74 and 84) — check the live ECP for the trigger conditions.

Exam format and permitted materials

All three are paper-based, invigilated and **closed book**, with the same calculator rule: **Casio FX82 series only**, or a UQ-approved and labelled calculator.

	In-semester (20%)	Lab exam (10%)	Final (50%)
Time	10 min perusal + **90 min**	10 min + **60 min**	10 min + **120 min**
Format	Multiple choice, individual	Practical/demonstration	MCQ, short answer, numerical, code writing
Covers	Weeks **1–5** — lectures, labs, homework/quizzes and readings	Lectures and practicals, weeks **1–5**	Everything (L01–L08)

	In-semester (20%)	Lab exam (10%)	Final (50%)
Extra	**Bring an HB or 2B pencil and an eraser**	**Lab kit permitted**	**One A4 sheet of handwritten or typed notes, double sided, is permitted.** AI not permitted

The lab exam runs **during your scheduled lab session (Mon–Fri) in week 7**, in a time slot announced on Blackboard.

Two **exam review sessions** run before the final: **Mon 2 November** and **Wed 4 November 2026, 10am–12pm, in 63-358**.

Late, extensions, deferrals

- **Assignment**: 10% of the maximum mark deducted **per 24 hours, up to 7 days**; after 7 days, a mark of **0**. Maximum extension **28 days**, in multiples of 24 hours.
- **Demo**: an assignment extension does **not** extend the demo — **apply separately**. Maximum extension 7 days, with one reschedule permitted via a my.UQ extension request. **No late submission accepted; 100% late penalty.** Non-attendance means 0 for the whole assignment.
- **Supplementary assessment** is available. It may be an exam, assignment, oral or combination at the coordinator's discretion, and you need **50% of the supplementary assessment** to pass (grade 3S4).

Resources

- Course Blackboard site: announcements, readings, notes, software downloads.
- Discussion Board (Edstem) for technical questions.
- Admin or content questions: use consultation hours, or email csse2010@eecs.uq.edu.au to book an appointment.

Lectures

Week 1

- 2026-07-27-course-intro-bits-bytes-binary[5] — Lecture 1

[5] courses/csse2010/2026-07-27-course-intro-bits-bytes-binary.pdf

- 2026-07-30-intro-to-logic-gates[6] — Lecture 2

Week 2

- 2026-08-04-binary-arithmetic[7] — Lecture 3
- 2026-08-06-combinational-logic[8] — Lecture 4

Week 3

- 2026-08-10-introduction-to-sequential-circuits[9] — Lecture 5
- 2026-08-13-shift-registers[10] — Lecture 6

Week 4

- 2026-08-17-counters[11] — Lecture 7
- 2026-08-20-state-machines[12] — Lecture 8

Week 5

- 2026-08-24-alu-and-control-unit[13] — Lecture 9
- 2026-08-27-avr-introduction[14] — Lecture 10

Week 6

- 2026-08-31-instruction-set-architecture-part-1[15] — Lecture 11
- 2026-09-03-instruction-set-architecture-part-2[16] — Lecture 12

Week 7

- 2026-09-07-introduction-to-embedded-c-programming[17] — Lecture 13
- 2026-09-10-more-c-programming-on-the-avr[18] — Lecture 14

Week 8

- 2026-09-14-flow-of-control[19] — Lecture 15

[6] courses/csse2010/2026-07-30-intro-to-logic-gates.pdf
[7] courses/csse2010/2026-08-04-binary-arithmetic.pdf
[8] courses/csse2010/2026-08-06-combinational-logic.pdf
[9] courses/csse2010/2026-08-10-introduction-to-sequential-circuits.pdf
[10] courses/csse2010/2026-08-13-shift-registers.pdf
[11] courses/csse2010/2026-08-17-counters.pdf
[12] courses/csse2010/2026-08-20-state-machines.pdf
[13] courses/csse2010/2026-08-24-alu-and-control-unit.pdf
[14] courses/csse2010/2026-08-27-avr-introduction.pdf
[15] courses/csse2010/2026-08-31-instruction-set-architecture-part-1.pdf
[16] courses/csse2010/2026-09-03-instruction-set-architecture-part-2.pdf
[17] courses/csse2010/2026-09-07-introduction-to-embedded-c-programming.pdf
[18] courses/csse2010/2026-09-10-more-c-programming-on-the-avr.pdf
[19] courses/csse2010/2026-09-14-flow-of-control.pdf

- 2026-09-17-pulse-width-modulation[20] — Lecture 16

Week 9

- 2026-09-21-interrupts[21] — Lecture 17
- 2026-09-24-serial-io[22] — Lecture 18

Week 10

- 2026-10-05-more-io[23] — Lecture 19
- 2026-10-08-assembly-process[24] — Lecture 20

  Monday 5 October was a public holiday (King's Birthday), so Lecture 19 was released as a recording.

Week 11

- 2026-10-15-compilation-and-linking[25] — Lecture 22

  Lecture 21 (Monday 12 October) was the assignment brief and has no note.

Week 12

- 2026-10-19-floating-point-numbers[26] — Lecture 23
- 2026-10-22-memory-disks-and-files[27] — Lecture 24

Exercises

Two prac sessions run per week (P2 Mon–Tue, P1 Thu–Fri, with P1 only in week 1), so **lab numbers run ahead of week numbers** — labs 2 and 3 are both week 2, labs 4 and 5 both week 3, and so on.

Week	Labs
1	week1-lab-number-representations[28] (Lab 1) · week-one-exercises[29]

[20] courses/csse2010/2026-09-17-pulse-width-modulation.pdf
[21] courses/csse2010/2026-09-21-interrupts.pdf
[22] courses/csse2010/2026-09-24-serial-io.pdf
[23] courses/csse2010/2026-10-05-more-io.pdf
[24] courses/csse2010/2026-10-08-assembly-process.pdf
[25] courses/csse2010/2026-10-15-compilation-and-linking.pdf
[26] courses/csse2010/2026-10-19-floating-point-numbers.pdf
[27] courses/csse2010/2026-10-22-memory-disks-and-files.pdf
[28] courses/csse2010/week1-lab-number-representations.pdf
[29] courses/csse2010/week-one-exercises.pdf

Week	Labs
2	week2-lab-majority-function-circuit-design[30] (Lab 2) · week2-lab-binary-arithmetic[31] (Lab 3)
3	week3-lab-lab-4-pre-work[32] (Lab 4) · week3-lab-logisim-and-flip-flops[33] (Lab 5)
4	week4-lab-sequential-circuits-shift-registers[34] (Lab 6) · week4-lab-lab-7-exercises[35] (Lab 7)
5	week5-lab-state-machines[36] (Lab 8) · week5-lab-avr-assembly-one[37] (Lab 9)
6	week6-lab-avr-io-ports[38] (Lab 10)
7	*no labs — lab exam week*
8	week8-lab-avr-c-programming[39] (Lab 11) · week8-lab-avr-timers[40] (Lab 12)
9	week9-lab-avr-pwm[41] (Lab 13) · week9-lab-avr-interrupts[42] (Lab 14)
10	Lab 15 (serial comms, no note) · week10-lab-adc-and-spi[43] (Lab 16)
11 onward	*no labs — sessions are for assignment work*

Also: binary-conversion-drill[44] — self-paced drill.

[30] courses/csse2010/week2-lab-majority-function-circuit-design.pdf
[31] courses/csse2010/week2-lab-binary-arithmetic.pdf
[32] courses/csse2010/week3-lab-lab-4-pre-work.pdf
[33] courses/csse2010/week3-lab-logisim-and-flip-flops.pdf
[34] courses/csse2010/week4-lab-sequential-circuits-shift-registers.pdf
[35] courses/csse2010/week4-lab-lab-7-exercises.pdf
[36] courses/csse2010/week5-lab-state-machines.pdf
[37] courses/csse2010/week5-lab-avr-assembly-one.pdf
[38] courses/csse2010/week6-lab-avr-io-ports.pdf
[39] courses/csse2010/week8-lab-avr-c-programming.pdf
[40] courses/csse2010/week8-lab-avr-timers.pdf
[41] courses/csse2010/week9-lab-avr-pwm.pdf
[42] courses/csse2010/week9-lab-avr-interrupts.pdf
[43] courses/csse2010/week10-lab-adc-and-spi.pdf
[44] courses/csse2010/binary-conversion-drill.pdf

Sample mid-semester exams

Six past papers with worked answers: csse2010-sample-midsem-1[45] · csse2010-sample-midsem-2[46] · csse2010-sample-midsem-3[47] · csse2010-sample-midsem-4[48] · csse2010-sample-midsem-5[49] · csse2010-sample-midsem-6[50]

> These are **2020** papers, sat under *open book* conditions. The 2026 in-semester exam is **closed book with no written materials permitted**, multiple choice, 90 minutes, and needs an HB or 2B pencil. Practise the content, not the conditions.

Reference notes

- binary-number-representations[51] · logic-gates[52] · boolean-algebra[53] · combinational-logic-blocks[54]
- sequential-circuits[55] · flip-flops-and-latches[56] · shift-registers[57] · counters[58] · state-machines[59]
- timing-diagrams[60] · circuit-schematics[61] · device-pinouts[62] · io-board[63]
- cpu-organisation[64] · arithmetic-logic-unit[65] · fetch-decode-execute-cycle[66]
- instruction-set-architecture[67] · memory-models[68] · addressing-modes[69]

[45] courses/csse2010/csse2010-sample-midsem-1.pdf
[46] courses/csse2010/csse2010-sample-midsem-2.pdf
[47] courses/csse2010/csse2010-sample-midsem-3.pdf
[48] courses/csse2010/csse2010-sample-midsem-4.pdf
[49] courses/csse2010/csse2010-sample-midsem-5.pdf
[50] courses/csse2010/csse2010-sample-midsem-6.pdf
[51] courses/csse2010/binary-number-representations.pdf
[52] courses/csse2010/logic-gates.pdf
[53] courses/csse2010/boolean-algebra.pdf
[54] courses/csse2010/combinational-logic-blocks.pdf
[55] courses/csse2010/sequential-circuits.pdf
[56] courses/csse2010/flip-flops-and-latches.pdf
[57] courses/csse2010/shift-registers.pdf
[58] courses/csse2010/counters.pdf
[59] courses/csse2010/state-machines.pdf
[60] courses/csse2010/timing-diagrams.pdf
[61] courses/csse2010/circuit-schematics.pdf
[62] courses/csse2010/device-pinouts.pdf
[63] courses/csse2010/io-board.pdf
[64] courses/csse2010/cpu-organisation.pdf
[65] courses/csse2010/arithmetic-logic-unit.pdf
[66] courses/csse2010/fetch-decode-execute-cycle.pdf
[67] courses/csse2010/instruction-set-architecture.pdf
[68] courses/csse2010/memory-models.pdf
[69] courses/csse2010/addressing-modes.pdf

- avr-architecture[70] · avr-instructions[71] · avr-io-ports[72] · avr-program-structure[73] · bit-masking[74]
- c-language-basics[75] · c-operators[76] · c-control-flow[77] · c-pointers-and-arrays[78] · c-preprocessor-and-avr-headers[79] · seven-segment-display[80]
- avr-branches-and-jumps[81] · procedures-and-the-stack[82] · avr-timers-and-counters[83] · pulse-width-modulation[84]
- polling-and-interrupts[85] · avr-interrupts[86] · software-timers[87]
- serial-communication[88] · avr-usart[89] · avr-spi[90] · analog-to-digital-conversion[91]
- assembly-process[92] · compilation-and-linking[93]
- fixed-point-numbers[94] · floating-point-numbers[95]
- memory-technologies[96] · memory-hierarchy-and-cache[97] · magnetic-disks[98] · file-systems[99]

[70] courses/csse2010/avr-architecture.pdf
[71] courses/csse2010/avr-instructions.pdf
[72] courses/csse2010/avr-io-ports.pdf
[73] courses/csse2010/avr-program-structure.pdf
[74] courses/csse2010/bit-masking.pdf
[75] courses/csse2010/c-language-basics.pdf
[76] courses/csse2010/c-operators.pdf
[77] courses/csse2010/c-control-flow.pdf
[78] courses/csse2010/c-pointers-and-arrays.pdf
[79] courses/csse2010/c-preprocessor-and-avr-headers.pdf
[80] courses/csse2010/seven-segment-display.pdf
[81] courses/csse2010/avr-branches-and-jumps.pdf
[82] courses/csse2010/procedures-and-the-stack.pdf
[83] courses/csse2010/avr-timers-and-counters.pdf
[84] courses/csse2010/pulse-width-modulation.pdf
[85] courses/csse2010/polling-and-interrupts.pdf
[86] courses/csse2010/avr-interrupts.pdf
[87] courses/csse2010/software-timers.pdf
[88] courses/csse2010/serial-communication.pdf
[89] courses/csse2010/avr-usart.pdf
[90] courses/csse2010/avr-spi.pdf
[91] courses/csse2010/analog-to-digital-conversion.pdf
[92] courses/csse2010/assembly-process.pdf
[93] courses/csse2010/compilation-and-linking.pdf
[94] courses/csse2010/fixed-point-numbers.pdf
[95] courses/csse2010/floating-point-numbers.pdf
[96] courses/csse2010/memory-technologies.pdf
[97] courses/csse2010/memory-hierarchy-and-cache.pdf
[98] courses/csse2010/magnetic-disks.pdf
[99] courses/csse2010/file-systems.pdf