Lancaster University

Topic 1:
SCC.131 Module introduction

Course Aims

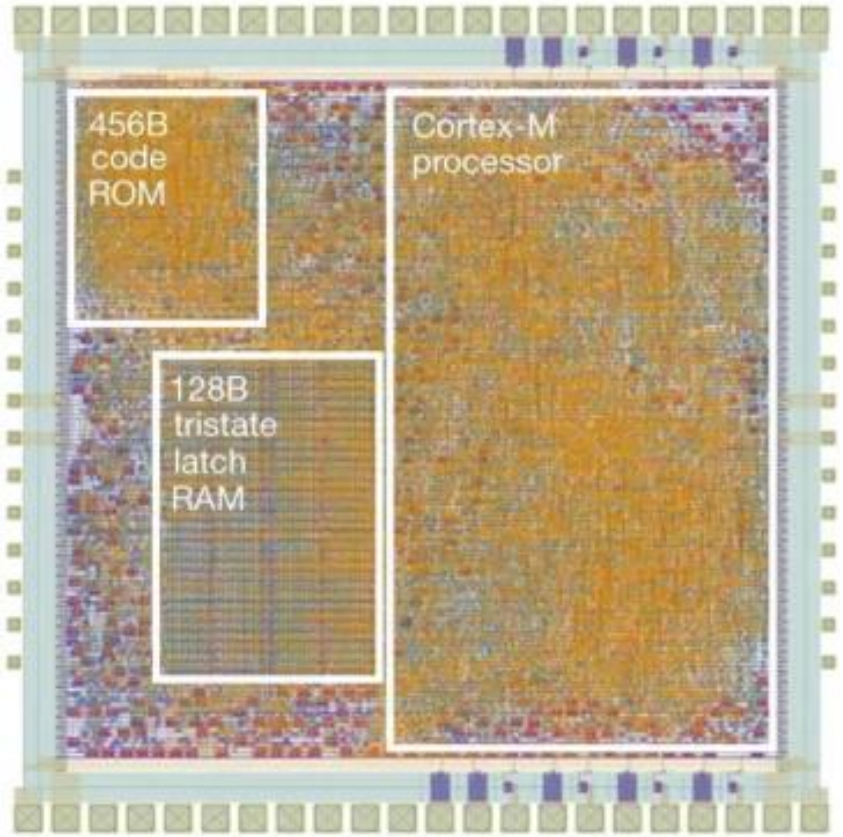
Lancaster University

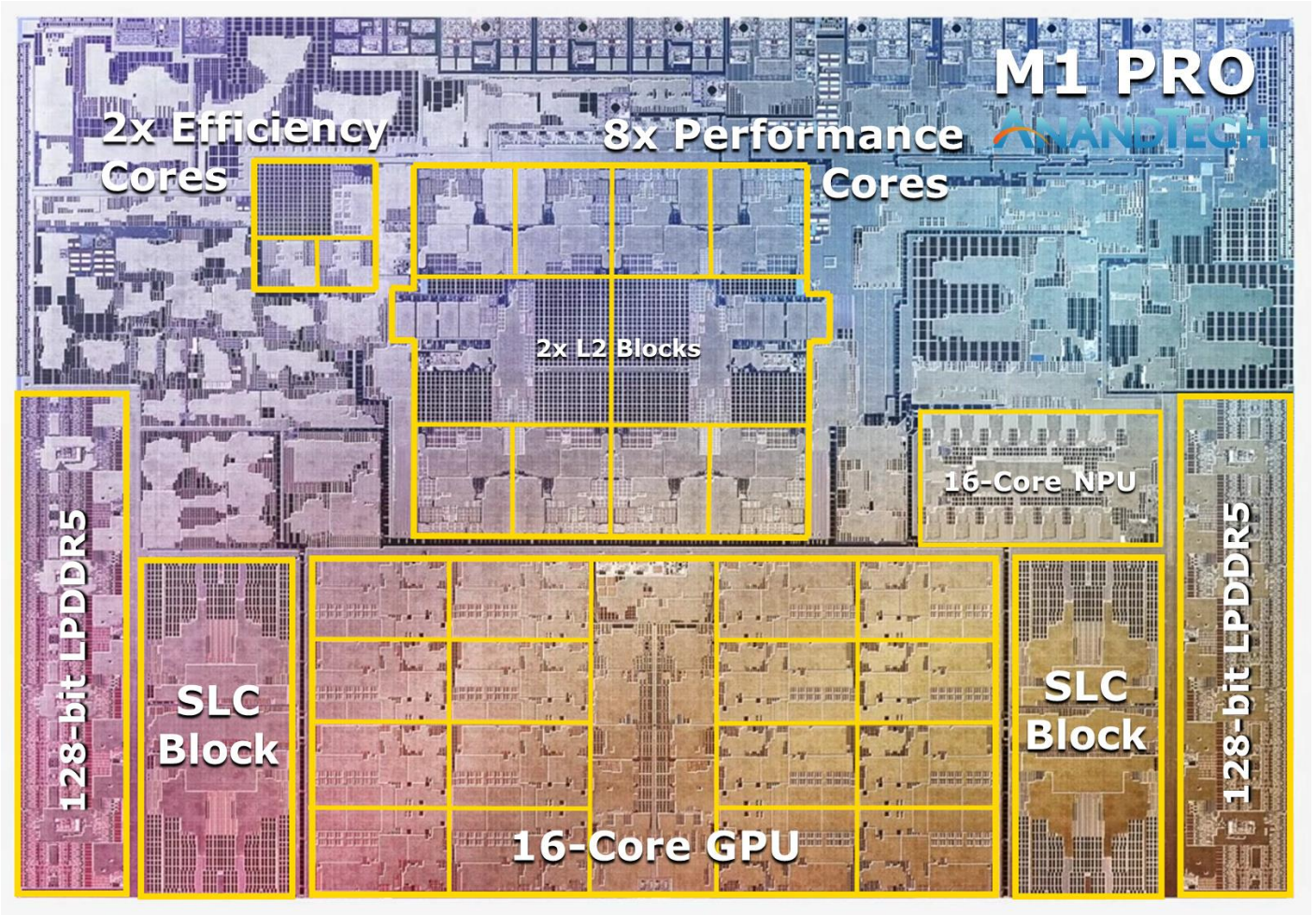
- To understand the fundamentals of digital systems:
  - Understanding hardware, from fundamental concepts and components to whole computer systems.
  - Understand how hardware and software interact.
  - Understanding how to program and debug at software levels that are “close to the machine”.
- This is an introductory module,
  - You might have seen some bits before (e.g. A-level, etc.)
  - It is crucial that you attend lecture slots and labs!
- Reduced participation is corelated with reduced performance.

What is common between the two CPUs?

Lancaster University

ARM Cortex-M die

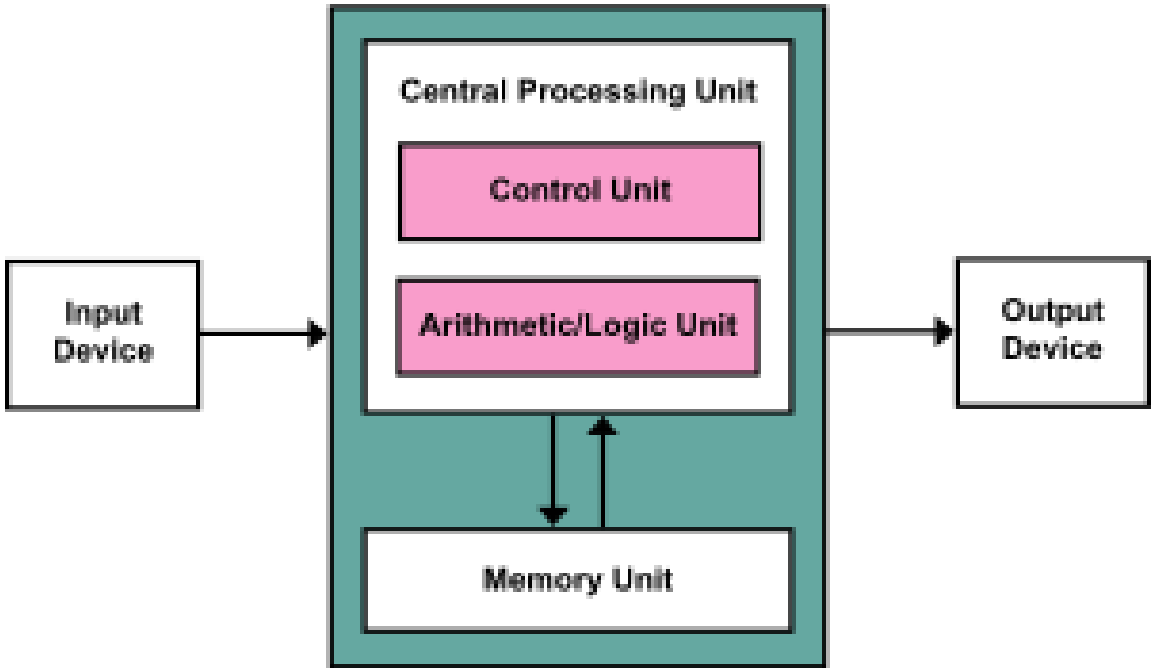


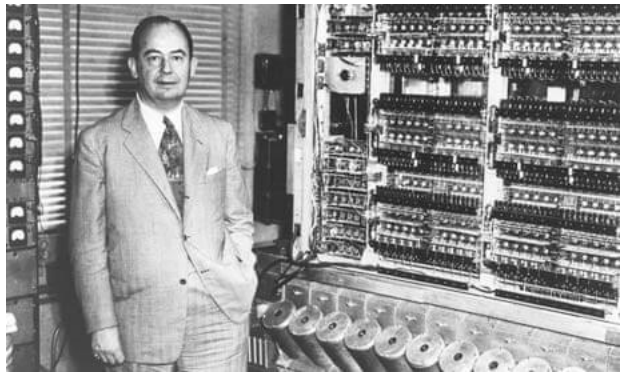


Apple M1 die

Von Neuman Architecture

Lancaster University

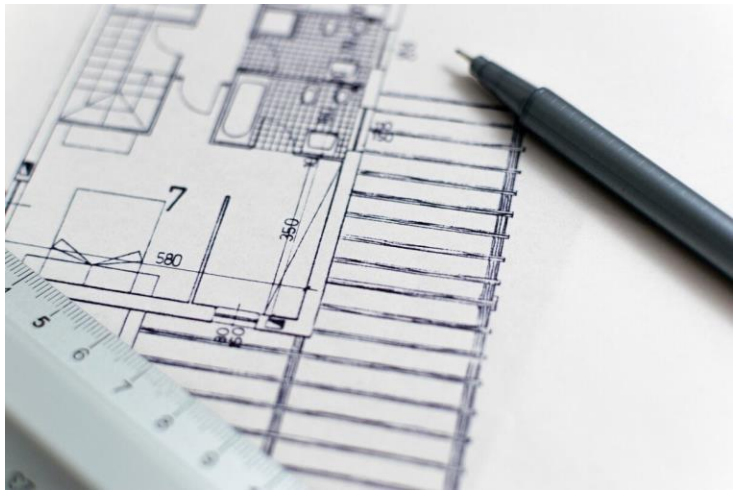


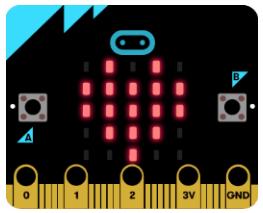


What is Computer Architecture?

Lancaster University

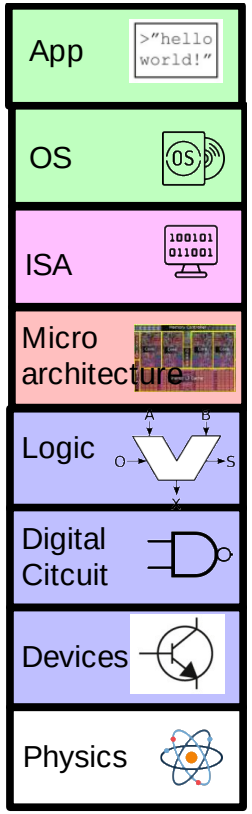
- *Architecture: The science of putting together building materials to produce aesthetically pleasing buildings.*
  - Material: bricks, glass, concrete …
  - Buildings: house, office, school …
  - Constraints: size, time, cost, health…





- *Computer Architecture: The science of creating computers, by putting together hardware components.*
  - *Hardware components: circuits, gates, chips …*
  - *Computers: desktop, server, mobile phone …*
  - *Constraints: performance, energy, cost …*

Module organization



Week 20-25 (Dr. Dempster): Systems Programming

Week 13-19 (Dr. Rotsos): ARM Assembly

Week 6-12 (Dr. Chatzigeorgiou): Microbit

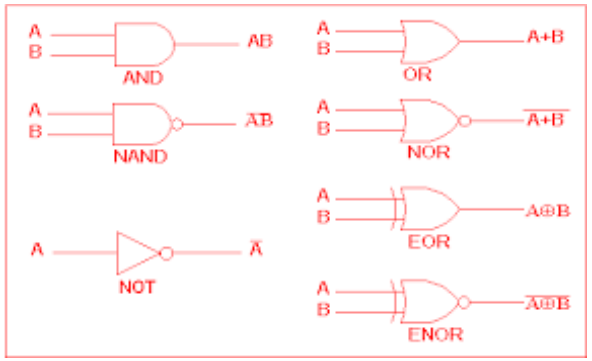
Week 1-6 (Prof. Ni): *Computer Architecture*

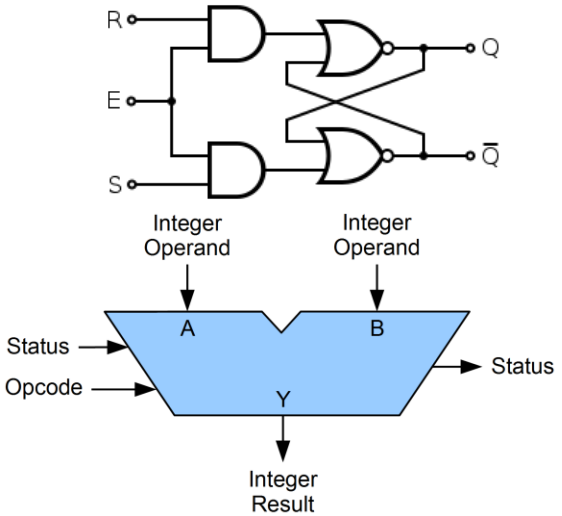
- Assembly & C
- Advanced C
- Networking Sockets
- Assembly programming
- Memory
- Interrupts / IO
- Debugging intro
- Physical computing
- Compilation
- Number systems
- Circuit and logic
- Computer Architecture Theory/ISA

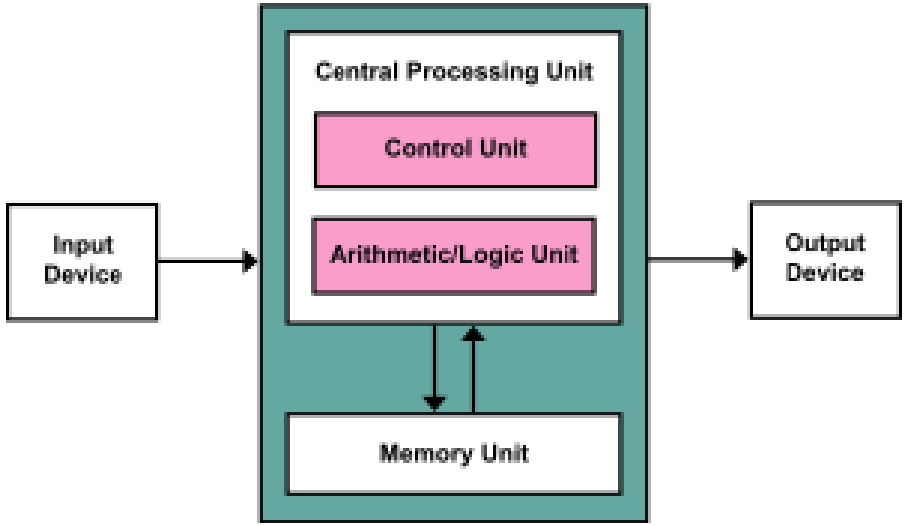
Computer Architecture

Lancaster University

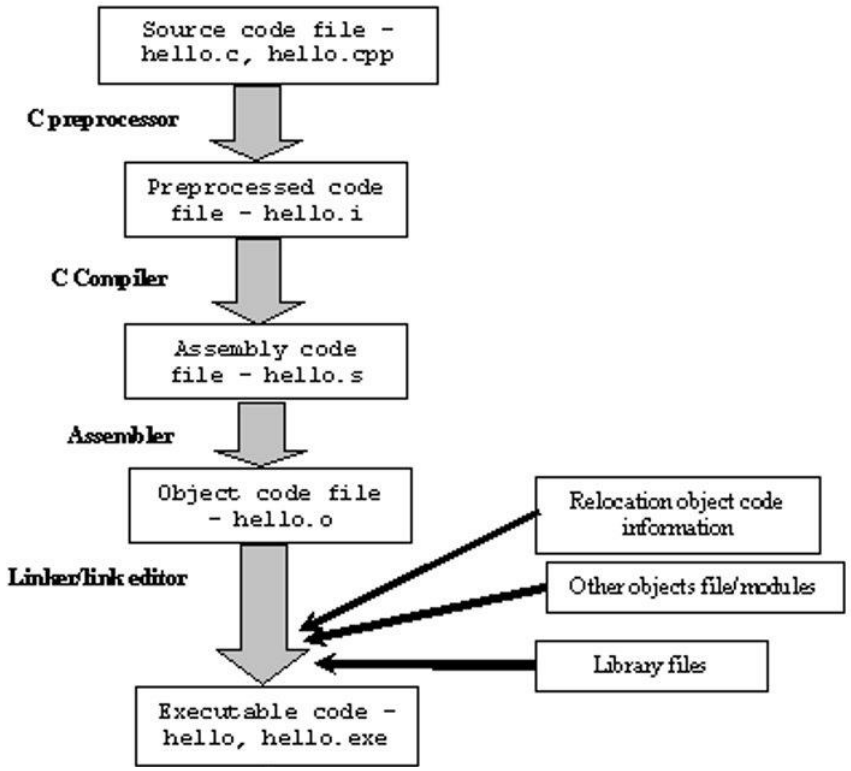
Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00000660	48	65	6C	6C	6F	2C	20	77	6F	72	6C	64	2E	36	34	30	Hello, world.640
00000670	30	39	30	39	31	36	35	30	30	30	31	39	36	32	33	0D	090916500019623.
00000680	0A	3A	31	30	31	45	35	30	30	30	39	30	39	33	36	35	.:101E5000909365
00000690	30	30	38	30	39	33	36	34	30	30	32	46	35	46	33	46	00809364002F5F3F
000006A0	34	46	38	30	39	31	36	36	30	31	45	46	0D	0A	3A	31	4F80916601EF..:1
000006B0	54	68	69	73	20	69	73	20	61	20	68	65	78	61	64	65	This is a hexade
000006C0	63	69	6D	61	6C	20	74	75	74	6F	72	69	61	6C	21	46	cimal tutorial!F
000006D0	38	39	34	45	31	39	39	33	36	0D	0A	3A	31	30	31	45	894E19936..:101E
000006E0	37	30	30	30	00	01	02	03	04	05	06	07	08	09	0A	0B	7000............
000006F0	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	....
00000700	1C	1D	1E	1F	20	21	22	23	24	25	26	27	28	29	2A	2B	.... !"#$%&'()*+
00000710	2C	2D	2E	2F	30	31	32	33	34	35	36	37	38	39	3A	3B	,-./0123456789:;
00000720	3C	3D	3E	3F	40	41	42	43	44	45	46	47	48	49	4A	4B	<=>?@ABCDEFGHIJK
00000730	4C	4D	4E	4F	50	51	52	53	54	55	56	57	58	59	5A	5B	LMNOPQRSTUVWXYZ[
00000740	5C	5D	5E	5F	60	61	62	63	64	65	66	67	68	69	6A	6B	\]^_`abcdefghijk
00000750	6C	6D	6E	6F	70	71	72	73	74	75	76	77	78	79	7A	7B	lmnopqrstuvwxyz{
00000760	7C	7D	7E	7F	80	81	82	83	84	85	86	87	88	89	8A	8B	|}~.€.‚ƒ„…†‡ˆ‰Š‹
00000770	8C	8D	8E	8F	90	91	92	93	94	95	96	97	98	99	9A	9B	Œ.Ž..‘’“”•–—˜™š›
00000780	9C	9D	9E	9F	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	œ.žŸ ¡¢£¤¥¦§¨©ª«
00000790	AC	AD	AE	AF	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	¬.®¯°±²³´µ¶·¸¹º»
000007A0	BC	BD	BE	BF	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	¼½¾¿ÀÁÂÃÄÅÆÇÈÉÊË
000007B0	CC	CD	CE	CF	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	ÌÍÎÏÐÑÒÓÔÕÖ×ØÙÚÛ
000007C0	DC	DD	DE	DF	E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	ÜÝÞßàáâãäåæçèéêë
000007D0	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	ìíîïðñòóôõö÷øùúû
000007E0	FC	FD	FE	FF	33	39	43	0D	0A	3A	31	30	31	45	44	30	üýþÿ39C..:101ED0
000007F0	30	30	35	37	30	30	45	38	39	35	33	32	39	36	30	32	005700E895329602







Instruction Set Architecture




```
int main() {
        uBit.init();
        const uint8_t heart[] = { 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1,
1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0,
0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, }; // a cute heart image

        i(10, 5, heart);
        uBit.display.animate(i, 1000, 1);

}
```

```
 .syntax unified
    .global func

    .text
    .thumb_func
func:
@ ----------------
@ Two parameters are in registers r0 and r1

    adds r0, r0, r1      @ Add r0 and r1, result in r0

@ Result is now in register r0
@ ----------------
    bx lr            @ Return to the caller
```

How is this course taught (1)

Lancaster University

	8:00	8:30	9:00	9:30	10:00	10:30	11:00	11:30	12:00	12:30	13:00	13:30	14:00	14:30	15:00	15:30	16:00	16:30
Mon																		
Tue																	SCC.131/L01/01 Lecture FAR - Faraday LT 1-10, 11-20, 21-25	
Wed											SCC.131/L01/02 Lecture MAN - Mngt School LT15 WPA016 1-10, 11-20, 21-25							
Thu													SCC.131/L02/01 Lecture BLM - Bowland Main LT 1-10		SCC.131/L02/02 Lecture BLM - Bowland Main LT 1-10			
Fri																		

How is this course taught (2)

Module: SCCx1A: SCC Lab Block A [1]

Weeks:

	8:00	8:30	9:00	9:30	10:00	10:30	11:00	11:30	12:00	12:30	13:00	13:30	14:00	14:30	15:00	15:30	16:00	16:30	17:00	17:30
Mon					SCCx1A/P01/02 Practical SAT - Science & Technology B070 1-10										SCCx1A/P01/08 Practical SAT - Science & Technology B074 1-10					
					SCCx1A/P01/01 Practical SAT - Science & Technology B080 1-10										SCCx1A/P01/07 Practical SAT - Science & Technology B070 1-10					
Tue					SCCx1A/P01/06 Practical SAT - Science & Technology B070 1-10						SCCx1A/P01/10 Practical SAT - Science & Technology B074 1-10									
					SCCx1A/P01/05 Practical SAT - Science & Technology B076 1-10						SCCx1A/P01/11 Practical SAT - Science & Technology B076 1-10									
Wed					SCCx1A/P01/04 Practical SAT - Science & Technology B080 1-10															
					SCCx1A/P01/03 Practical SAT - Science & Technology B074 1-10															
Thu					SCCx1A/P01/09 Practical SAT - Science & Technology B074 1-10															
Fri																				

Practical Requirements

Lancaster University

1. Computer Architecture
   - Pen and paper exercises and quizzes.
2. Embedded Systems
   - C code for microbit.
   - https://scc-source.lancs.ac.uk/scc.Y1/scc.131/microbit-v2-samples
3. Assembly
   - ARM M0 assembly on microbit.
4. Systems Programming
   - Linux and libc examples on x86.
   - Lab machines/VMs.

Teaching team

Lancaster University

- Prof. Qiang Ni

- Dr Ioannis Chatzigeogriou

- Dr Charalampos (Haris) Rotsos

- Dr Paul Dempster

How is this course assessed

Lancaster University

- *Exam/Coursework split: 70%/30%*
- *Coursework:*
  - *Architecture Quiz: Week 5 (5%)*
  - *Architecture + Embedded Systems: Week 10 (5%)*
  - *Assembly + Debugging Quiz: Week 15 (5%)*
  - *Assembly: Week 20 (5%)*
  - *Programming project: Week 23 (10%)*
- *Exams: Week 28-30 (Online, moodle)*
- *Use your time wisely.*
- *Coursework is submitted online and checked for plagiarism.*

Practical Organization

Lancaster University

- Assignments are released on Moodle over the weekend of each week.
  - https://modules.lancaster.ac.uk/course/view.php?id=41307
- You are expected to spend some time before the start of the lab.
- Bring pen and paper in the first term.
- Your lab machine will have pre-installed all the tools required to complete a task.

- Use your lab time efficiently:
  - Take advantage of TAs and academics and ask questions.
  - If you finish early, ask us to give you more tasks.

What is Plagiarism?

Lancaster University

- Passing off someone else's work as your own, including:
  - Submitting (e.g.) code that someone else wrote
  - Paying for someone else to do it for you
  - Working on a piece of non-group work together as a group, and submitting it as individual work
  - Sharing of code that you then possibly adapt
- If you **give someone else your work**, you can also be called in for plagiarism
- Coursework submitted online is checked for plagiarism *automatically*

- *If you use github/scc-source.lancs.ac.uk repos these* ***need to be private (still plagiarism)***

What We Expect from You

Lancaster University

- Integrity (no plagiarism, no faking results) and effort (active learning):
  - Come to lectures
  - Go to labs (these are compulsory!)
  - Use our/the world's resources effectively
  - Take notes
  - Read around the subject/try things for yourself
  - Ask us questions in lectures and labs
  - Take notes (again, because the slides are not enough when you try to revise, really…!)
  - Plan your time and coursework carefully

  - Please avoid contacting TAs out of hours; Use labs to get help.

What You can Expect from Us

Lancaster University

- We'll do our best
  - To make all our lecture notes/videos available on moodle
  - To personally check the labs are running smoothly and the TAs are offering support
  - To arrange extra support if you've already tried the normal routes (book, web, forum, TAs)
    - FAST sessions every day
    - One-to-one bookable sessions on moodle page
  - To offer prompt feedback on coursework

How do I get help?

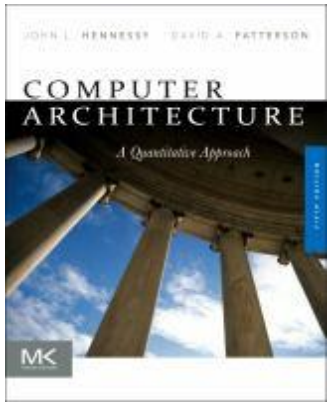
Lancaster University

- *With the **coursework** …*
  - Use the labs – that's what they are there for!
  - Join a FAST session
  - Ask the TAs: they are coursework experts
  - Use the forum on Moodle
  - Please avoid contacting TAs out of hours
- *With the **course** …*
  - Use the forum on Moodle (no code sharing), use Google or ask the TAs
  - Look at the course textbooks (see next slide)
  - Contact the lecturers

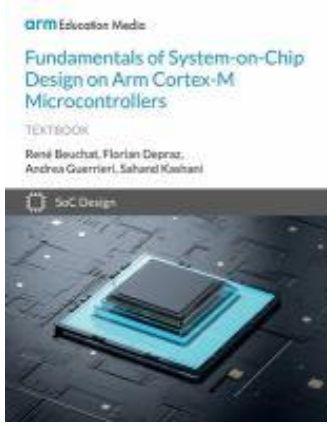
Course Textbooks (1)

Lancaster University

- https://eu.alma.exlibrisgroup.com/leganto/readinglist/lists/87115698570001221



Computer Architecture, Hennessy, Patterson
Good book that has added information for most topics discussed in the course.

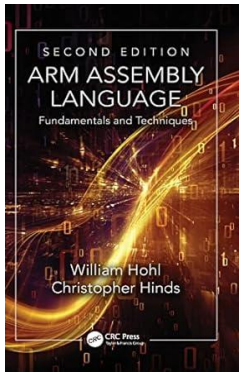


Fundamentals of System-on-Chip Design on Arm Cortex-M Microcontrollers, Beuchat, Depraz, Kashani, Guerrieri
Good book from the ARM academy explaing the architecture of the ARM M0

Course Textbooks (2)

Lancaster University

- https://eu.alma.exlibrisgroup.com/leganto/readinglist/lists/87115698570001221

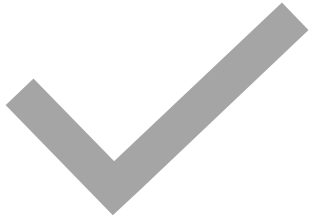


ARM assembly language: fundamentals and techniques, Hohl
All around cover of ARM assembly

Dive into Systems, Matthews, Newhall, Webb
Open-source book provide a top-down cover of several SCC.131 topics
https://diveintosystems.org/book/

Conclusions

Lancaster University



Introduction to module

Module Plan

Lab details

Assessment

Next Sessions

Architecture Introduction