컴퓨터 개념 및 실습

중간 고사 review

Before we get started…

- Plan for 4/16 (Mon)

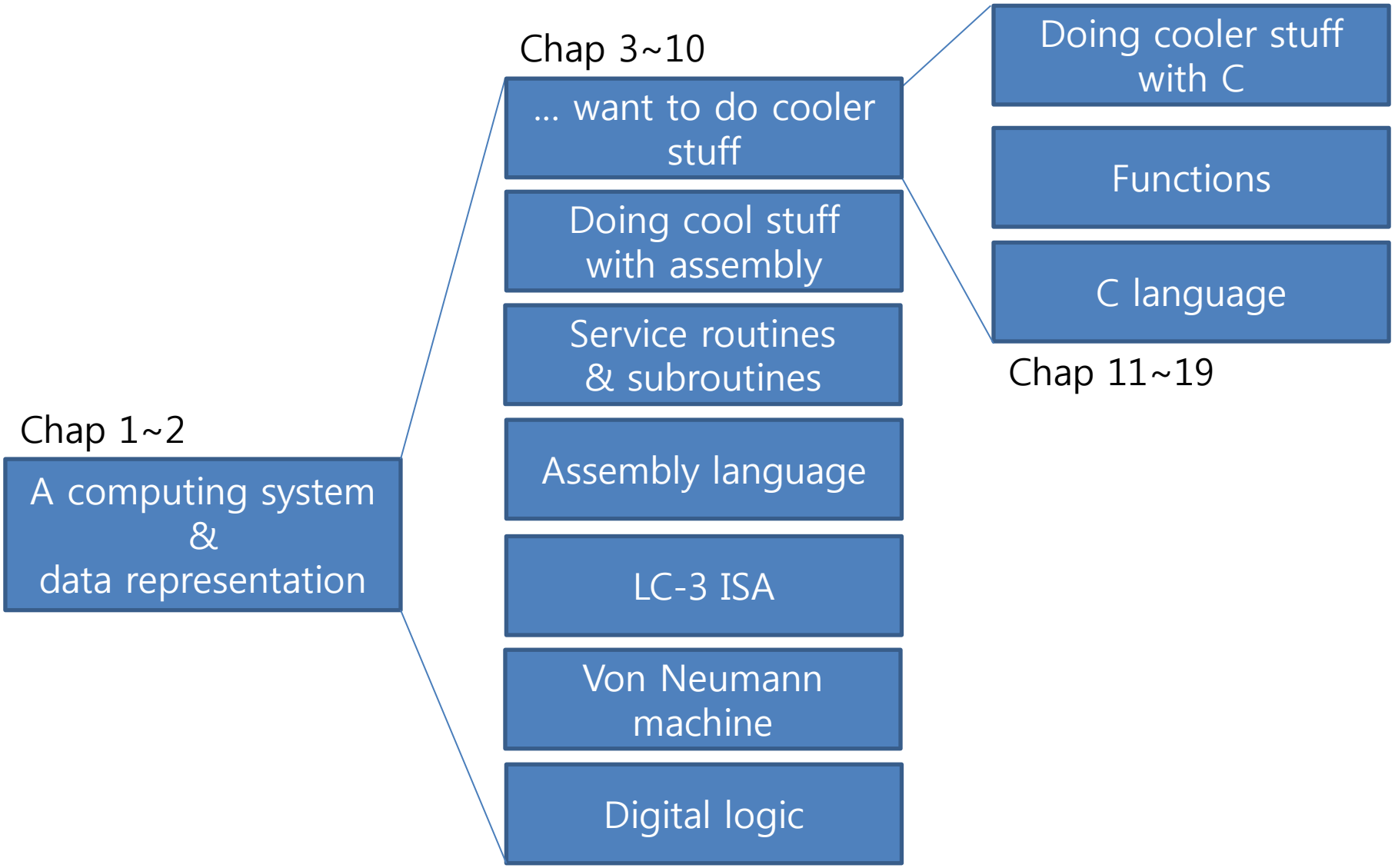
- Projects
  - Individual project
  - Group project

- Midterm reviews
  - What I think is important for each chapter
  - Questions I got via e-mail
  - Possibly going over more example exercises

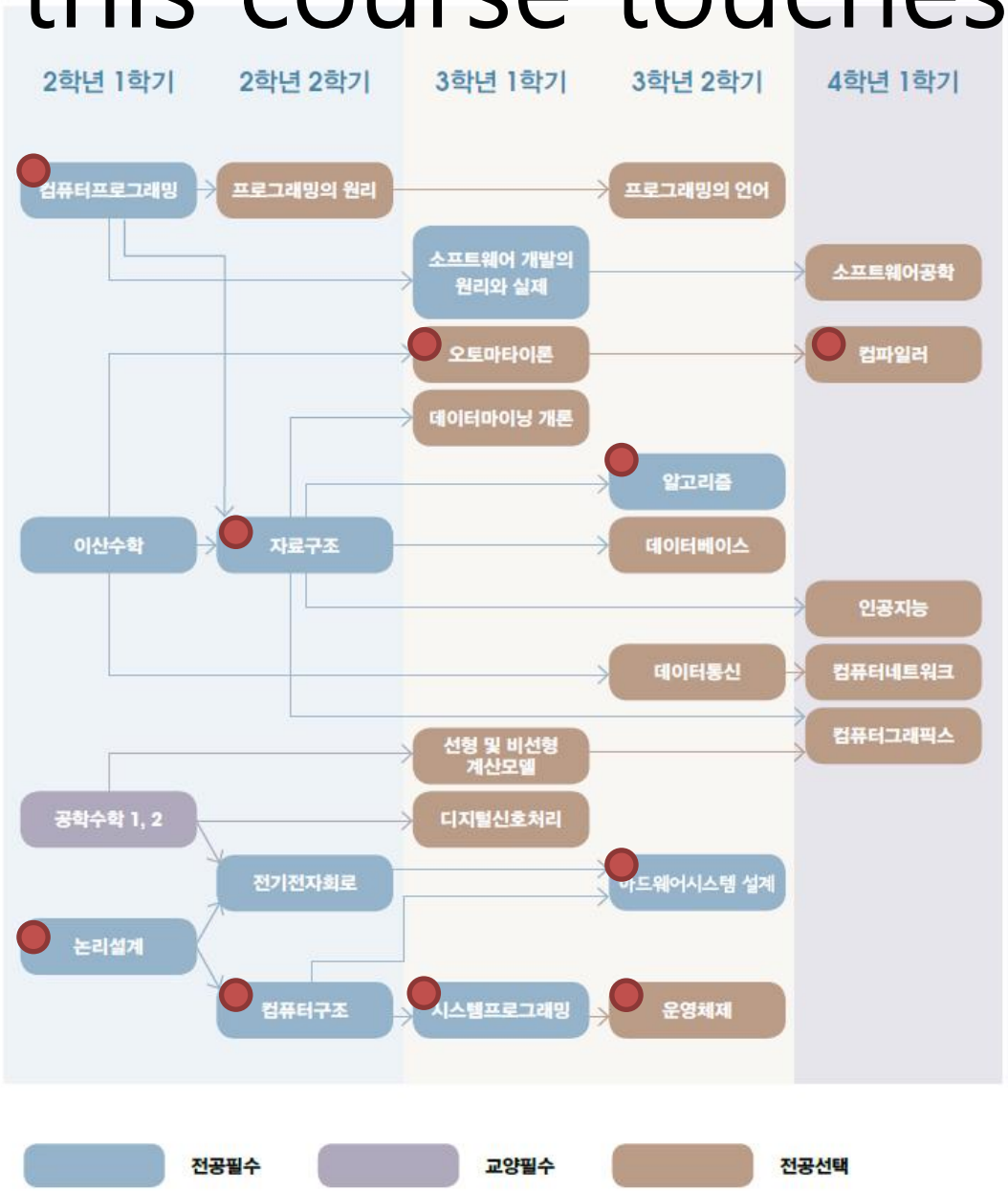
The entire course in a nutshell

- The underlying structure of a computer
  - Chap 1~Chap 10
- Programming in a high-level language
  - Chap 11~Chap 19

My interpretation of the textbook

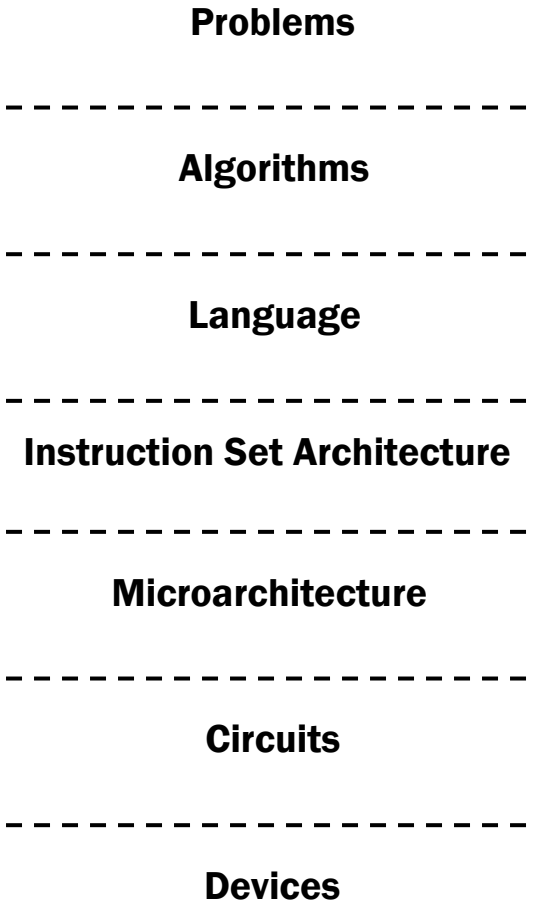


What this course touches upon



Important ideas from Chap. 1

- Abstraction
- HW vs. SW
- Universal computing device



* Fig. 1.6 in textbook

Important ideas from Chap 2.

- Integer representation
  - Sign-magnitude
  - 1's complement
  - 2's complement
- Operations
  - Arithmetic: add, subtract, sign-ext, overflow
  - Logical: and, or, not… nand, nor, xor, xnor
- Other data representations
  - Floating point
  - ASCII

Important ideas from Chap 3.

- Transistor
- Logic gates
  - Truth table
  - DeMorgan's Law, and other laws
- Combinational logic
  - Decoder
  - Mux
  - Adder
- Storage elements
  - Latch
  - Gated latch
  - Register
  - Memory
- Sequential logic
  - Flip-flop
  - Finite state machine

Important ideas from Chap 4.

- Von Neumann model
  - Memory
  - Processing unit
  - Control unit
  - Input
  - Output
- Inst. processing
  - Fetch inst.
  - Decode
  - Eval. addr.
  - Fetch operands
  - Execute
  - Store

Important ideas from Chap 5.

- LC-3 ISA
  - Memory, registers
  - Instructions
- LC-3 instructions
  - Operate
  - Data movement
  - Control
- Details in LC-3 ISA
  - Addressing modes
  - Condition codes
  - What the control unit does for each inst.

Important ideas from Chap 6.

- Problem solving
  - Step-wise refinement
  - Constructs: seq, cond, iter
- Debugging
  - What a debugger provides
  - How to debug: step, breakpoint, watchpoint
  - Types of mistakes: syntax, logic, data

Important ideas from Chap 7.

- Assembly language syntax
  - Instruction
  - Label
  - Comment
  - Pseudo-op
- Assembly process
  - Two-pass process
    - Symbol table
    - Location count

Important idea from Chap 8.

- I/O (device) ←→ CPU
  - Device register
  - Memory-mapped vs. special instructions
  - Sync vs. async
  - Polling vs. interrupt

- LC-3 computer's keyboard & monitor I/O

E-mail questions

- Bit vector, busyness, masking
- Exercise 2.38
- Exercise 2.39c
- Exercise 3.22
- Exercise 3.24
- Exercise 5.2

Exercise 2.38 (and 2.37)

MSB carry-in	M[3]	N[3]	S[3]	Signed Overflow?	Unsigned Overflow?
0	0	0	0	No	No
0	0	1	1	No	No
0	1	0	1	No	No
0	1	1	0	Yes	Yes
1	0	0	1	Yes	No
1	0	1	0	No	Yes
1	1	0	0	No	Yes
1	1	1	1	No	Yes

Signed overflow: (m[3]&n[3]&~s[3]) | (~m[3]&~n[3]&s[3])
Unsigned overflow: (m[3]&n[3]) | (~m[3]&n[3]&~s[3]) | (m[3]&~n[3]&~s[3])

Exercise 2.39c

- 3.1415927 in IEEE 754 floating point…

  = 2 * 1.57079635

  (in the form of 1.fraction * 2^1)

  →Sign = 0, Exponent = 128

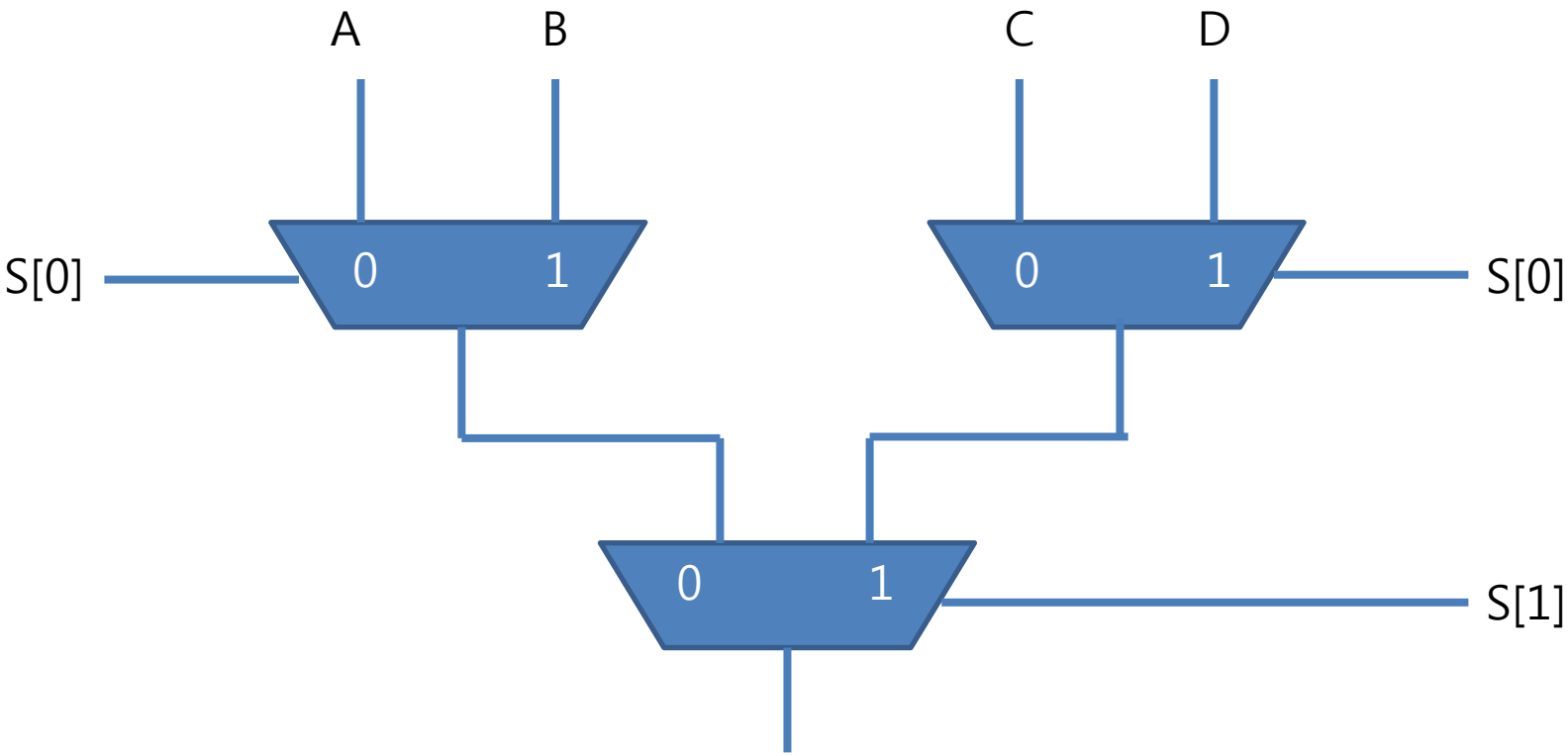
  Now need to express 0.57079635

  0.57079635

  = 0.5 + 0.0625 + 0.0078125 + ….

Exercise 3.22

- Implement 4-to-1 mux using 2-to-1 muxes



Exercise 3.24

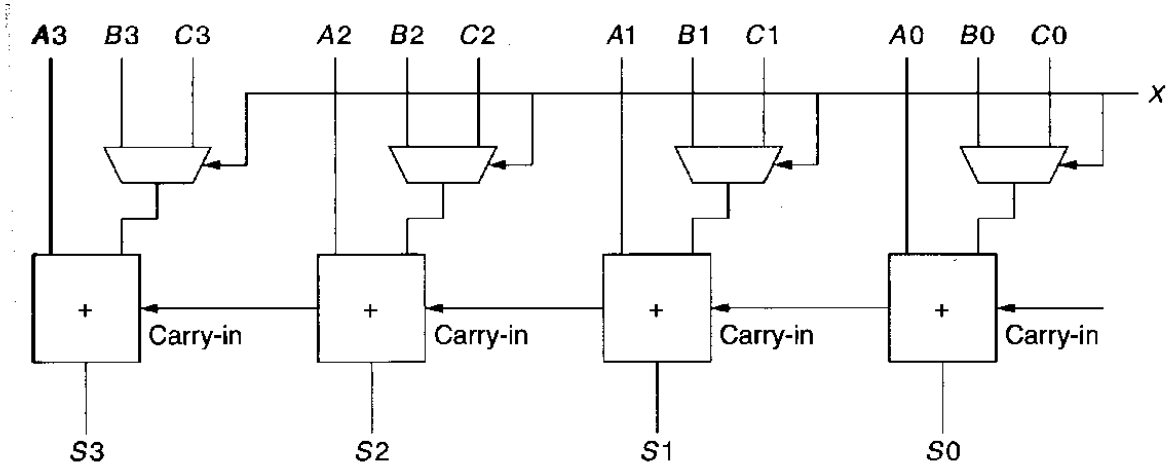


Figure 3.39 Diagram for Exercise 3.24

- A) S = A + B ; if X is 0
  S = A + C ; otherwise
- B) To implement adder/subtractor
  - Make C = ~B
  - Connect X to LSB carry-in

Exercise 5.2

- Memory's addressibility is 64 bits
  - MDR is 64 bit in size
  - It doesn't say anything about MAR size