Digital System Design
數位系統設計

Jean Shann 單智君
Dept. of Computer Science
NYCU
Fall 2022

J.J. Shann

Class & Office

＊ 每週
兩小時實體課程 +
一小時非同步線上課程

- Class:
  - 2-hour in-person/on-site class: Thur. 56 (EC015)
  - 1-hour asynchronous on-line class: Lecture videos
- Office:
  EC516, ext. 31832
  jjshann@cs.nctu.edu.tw (e3 mail system)
- On-line Office Hours:
  Wed. 2:00PM~3:00PM
  Google meet: https://meet.google.com/tgh-uyjr-fpr
  (or making appointments via e-mail)

Teaching Assistant & TA Hours

- 徐子瀚 教學助理
  - e-mail: hankshyu.cs08@nycu.edu.tw
  - On-line TA Hours:
    Tue. 6:30PM~8:30PM
    Google meet: https://meet.google.com/boi-pmmw-psq
    (or making appointments via e-mail)

校園實體上課防疫措施

(2022/09/02 updated)

- 各項課程依原排定授課方式進行。
- 實體上課課程若有學生屬於等候入境之境外生、檢疫、確診、居家隔離等無法到課者，請學生聯繫任課老師提供同步或非同步線上上課機制，以確保學習受教權。學生可透過E3系統課程內的寄信功能聯繫老師或助教。
- 教師及學生進行實體課程時，應全程配戴口罩，落實手部消毒，上課期間禁止飲食。
- 教室應依照相關防疫規定進行消毒及清潔，並保持室內通風良好。

Prerequisite

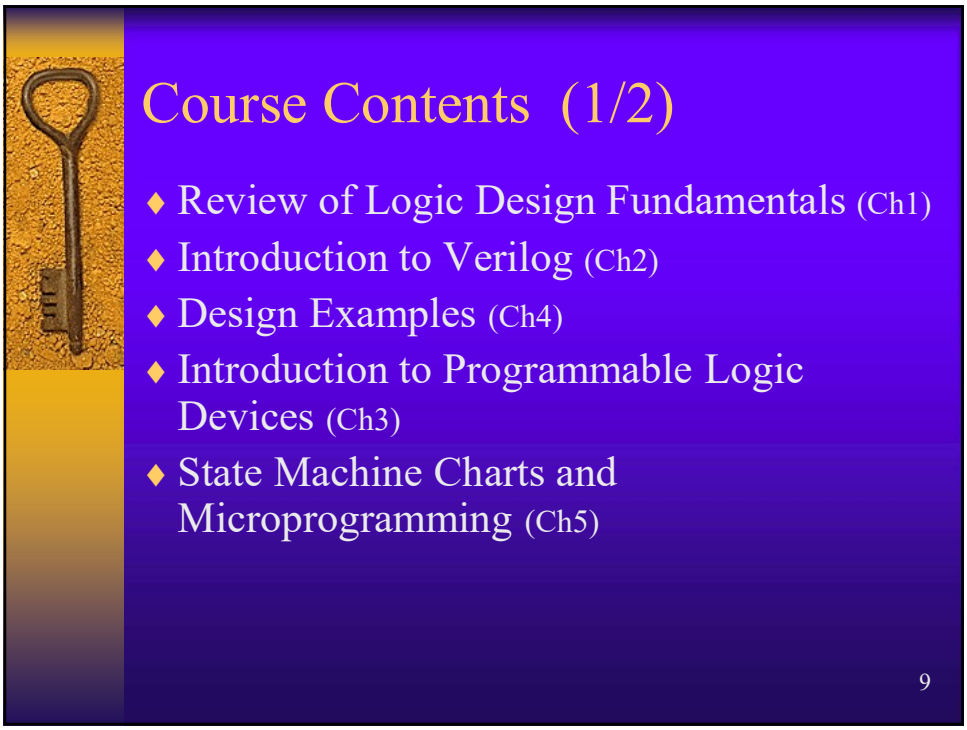
- Prerequisite:
  - Digital Circuits Design 數位電路設計
    (Logic Design 邏輯設計)
  - Hardware Description Language:
    Verilog HDL (or VHDL)

Textbook & References

- Textbook:

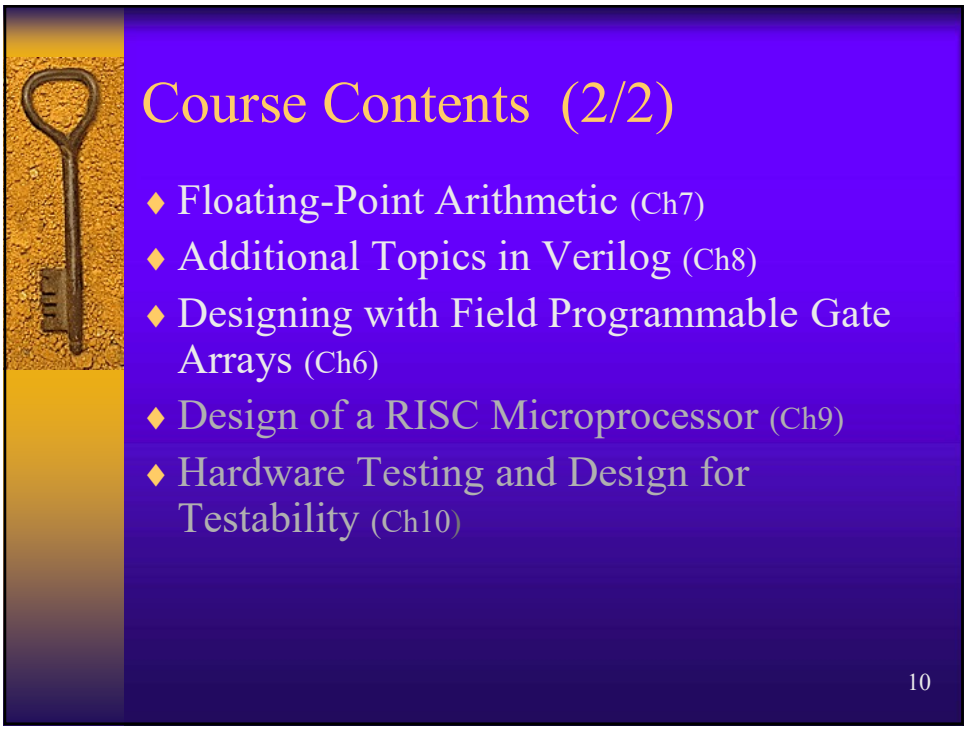
  Charles Roth, Lizy Kurian John, and Byeong Kil Lee, *Digital Systems Design Using Verilog*, International Edition, 2016, Cengage Learning.

- References:
  - M. Morris Mano, Charles R. Kime, and Tom Martin, *Logic and Computer Design Fundamentals*, 5th Edition, 2016, Pearson Prentice Hall.
  - M. Morris Mano and Michael D. Ciletti, *Digital Design*, 6th ed., 2019, Prentice Hall.
  - Charles H. Roth, Jr., and Larry L. Kinney, *Fundamentals of Logic Design*, 7th ed. (International Edition), 2014, Cengage Learning.



Course Contents (1/2)

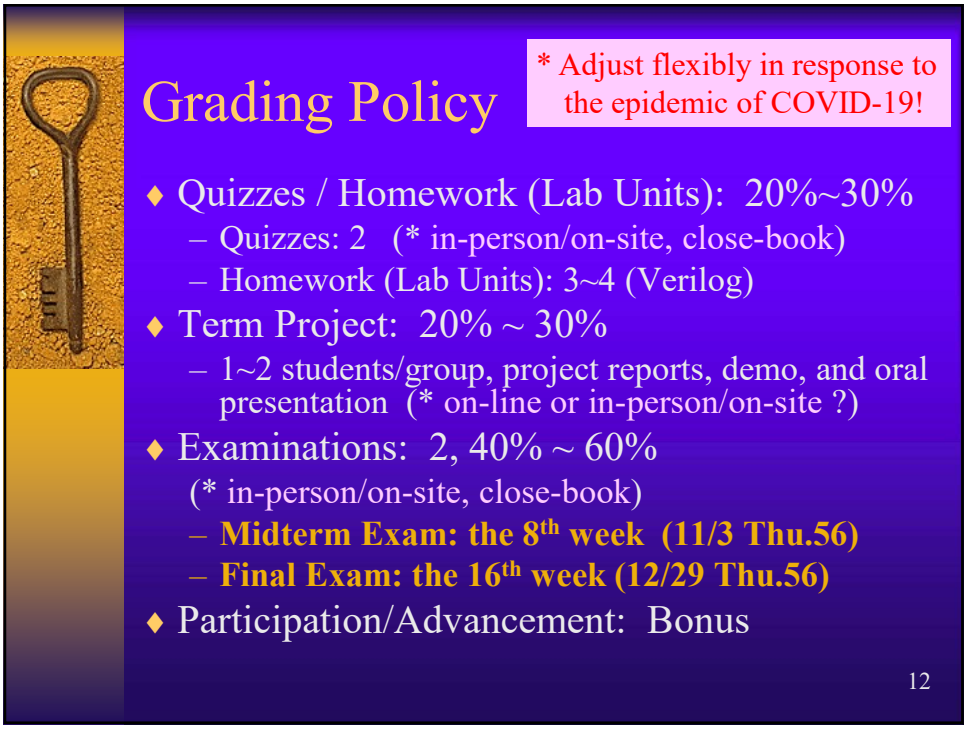
- Review of Logic Design Fundamentals (Ch1)
- Introduction to Verilog (Ch2)
- Design Examples (Ch4)
- Introduction to Programmable Logic Devices (Ch3)
- State Machine Charts and Microprogramming (Ch5)

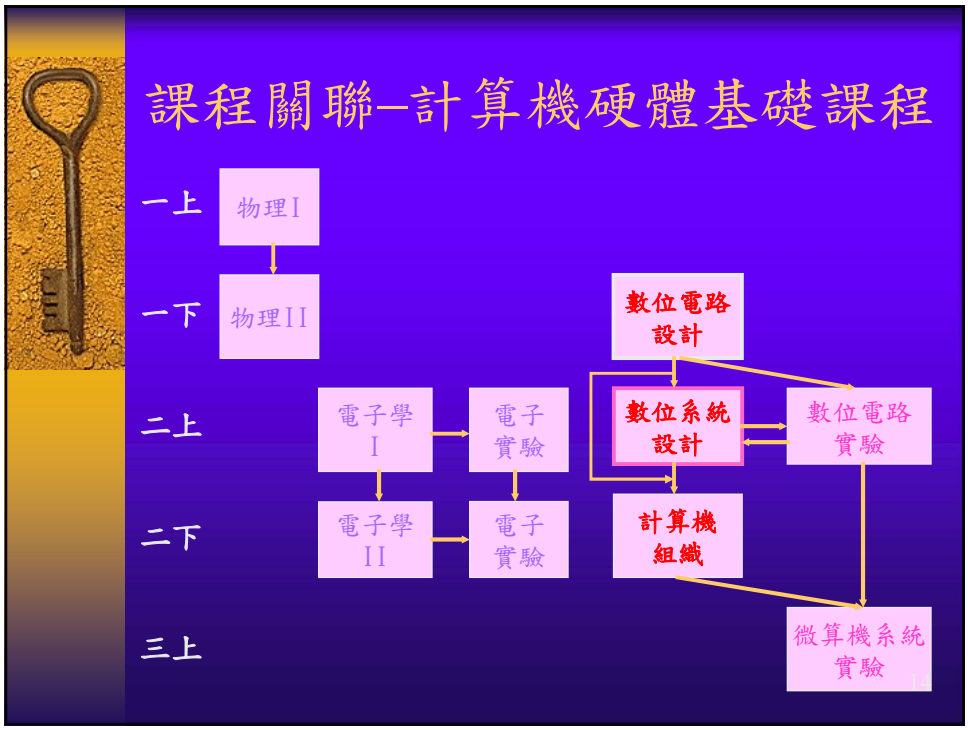


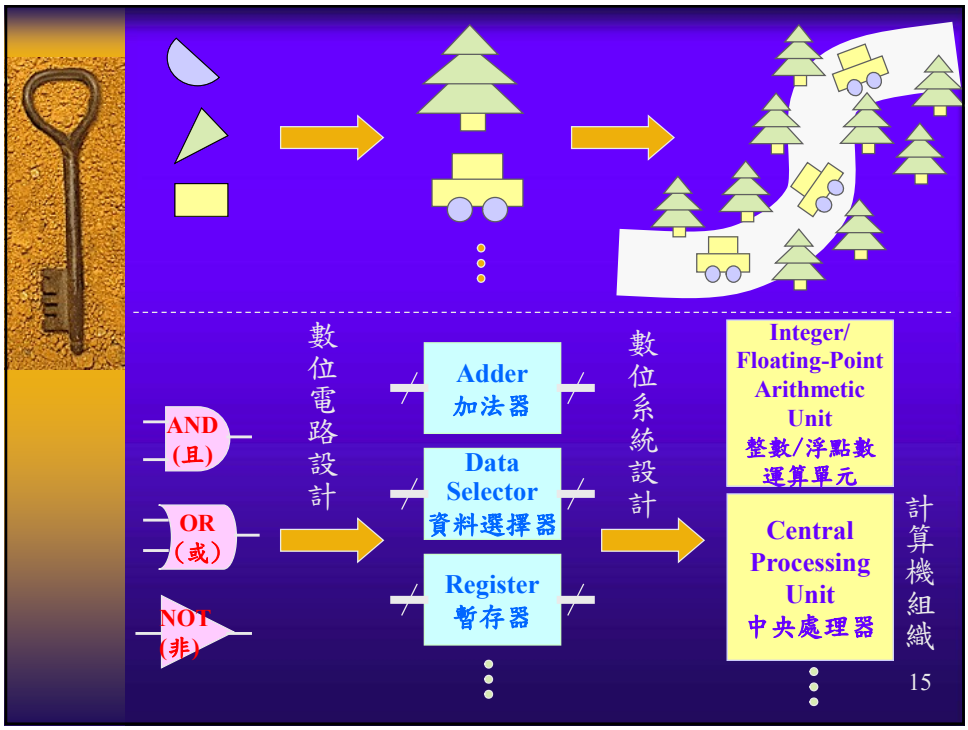
Course Contents (2/2)

- Floating-Point Arithmetic (Ch7)
- Additional Topics in Verilog (Ch8)
- Designing with Field Programmable Gate Arrays (Ch6)
- Design of a RISC Microprocessor (Ch9)
- Hardware Testing and Design for Testability (Ch10)


Course Information & Videos
♦ Course Information:
– announcements, slides, class video,…
– http://dcpc.nycu.edu.tw (e-Campus 網路教學平台)『數位系統設計 (Digital System Design)』 課程
11


Grading Policy
* Adjust flexibly in response to the epidemic of COVID-19!
♦ Quizzes / Homework (Lab Units): 20%~30%
– Quizzes: 2 (* in-person/on-site, close-book)
– Homework (Lab Units): 3~4 (Verilog)
♦ Term Project: 20% ~ 30%
– 1~2 students/group, project reports, demo, and oral presentation (* on-line or in-person/on-site ?)
♦ Examinations: 2, 40% ~ 60%
(* in-person/on-site, close-book)
– Midterm Exam: the 8th week (11/3 Thu.56)
– Final Exam: the 16th week (12/29 Thu.56)
♦ Participation/Advancement: Bonus
12


課程關聯–計算機硬體基礎課程
一上
物理I
一下
物理II
二上
電子學I
電子實驗
二下
電子學II
電子實驗
三上
數位電路設計
數位系統設計
計算機組織
數位電路實驗
微算機系統實驗


AND (且)
OR (或)
NOT (非)
數位電路設計
Adder 加法器
Data Selector 資料選擇器
Register 暫存器
數位系統設計
Integer/ Floating-Point Arithmetic Unit 整數/浮點數運算單元
Central Processing Unit 中央處理器
計算機組織
15

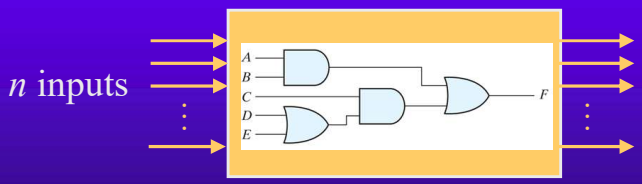
Review of Digital Circuits

- Combinational circuits
- Sequential circuits
  - Latches
  - Flip-Flops
  - Synchronous sequential circuits
  - Asynchronous sequential circuits

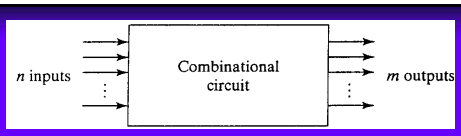
Combinational Circuits

- Combinational ckt: logic gates
  - It outputs at any time are determined from the present inputs. (no feedback paths or memory elements)

(inputs) $\Rightarrow$ (outputs)



- Examples:
  - Parallel adder, Encoder, Decoder, Multiplexer, …



- Analysis procedure:

  Logic diagram → Output Boolean functions,
  a truth table, or
  a verbal explanation of the ckt op

  1. Label all gate outputs that are a function of input variables w/ symbols.
  2. Determine the Boolean function for each gate output.
  3. Label the gates that are a function of input variables and previously labeled gates w/ other symbols.
  4. Find the Boolean functions for these gates.
  5. Repeat step 2 until the outputs of the ckt are obtained in terms of input variables.

- Design procedure:

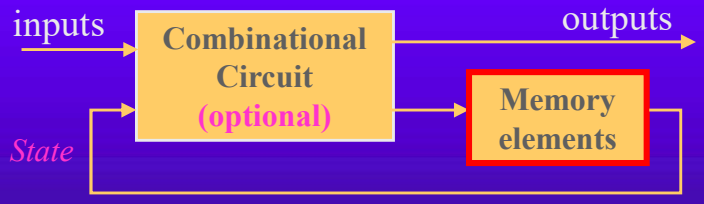
  Specification of the problem
  → Logic ckt diagram or a set of Boolean functions

  1. Specification: From the specifications of the ckt, determine the required # of inputs and outputs and assign a symbol to each.
  2. Formulation: Derive the truth table that defines the required relationship b/t inputs and outputs.
  3. Optimization:
     - Apply two-level and multiple-level optimization: Obtain the simplified Boolean functions for each output as a function of the input variables.
     - Draw a logic diagram for the resulting ckt using ANDs, ORs, and NOTs.
  4. Technology mapping: Transform the logic diagram to a new diagram using the available implementation technology.
  5. Verification: Verify the correctness of the design.

Sequential Circuits

- Sequential circuit:

  (inputs, present state) ⇒ (outputs, next state)

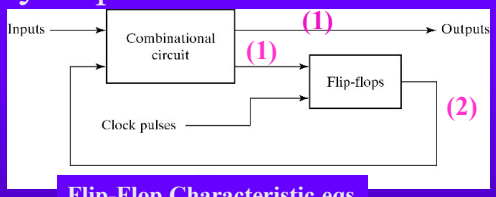


Memory elements: devices capable of storing binary information (*states*)

- Synchronous vs. Asynchronous
- Examples: Vendor machine, Serial adder, …

Synchronous Sequential Circuit

- Analysis procedure:



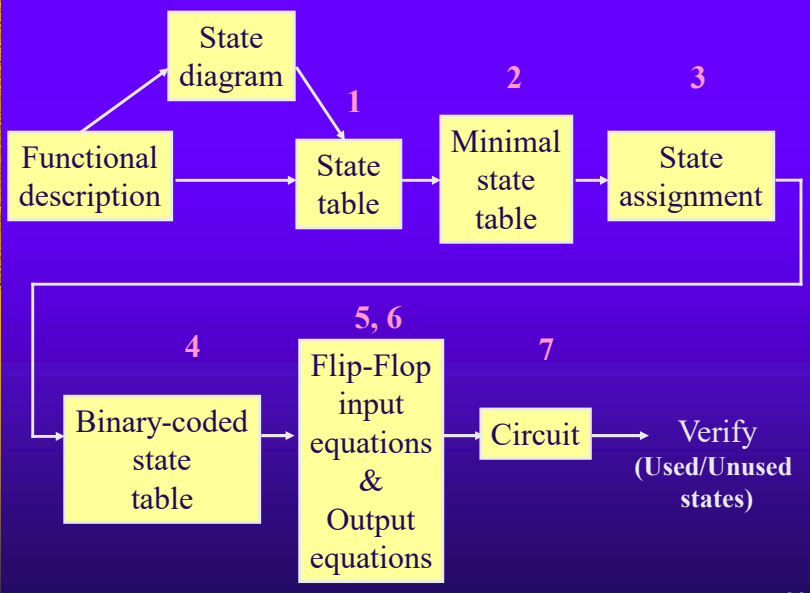
Flip-Flop Characteristic eqs

1 2 3 4

Circuit (sync seq) → Flip-flop input Equations → Next state equations → State table (Transition table) → State diagram

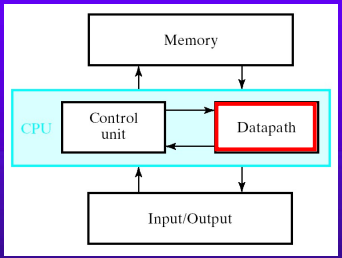
Output equations → State table (Transition table)

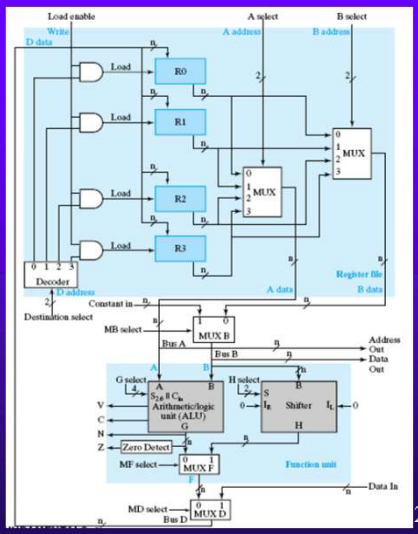
- Design procedure:
  1. From the word description and specifications of the desired operation, derive a state diagram or state table for the ckt.
  2. Reduce the # of states if necessary.
  3. Assign binary values to the states.
  4. Obtain the binary-coded state table.
  5. Choose the type of flip-flops to be used.
  6. Derive the simplified flip-flop input equations and output equations.
  7. Draw the logic diagram with flip-flops and combinational gates.
  8. Verify the correctness of the design.



An Example of Digital Systems

- Digital computer:





Revolution in Hardware Design

- Pervasive use of software tools to assist in the process of hardware design
  - Hardware description language (HDL)
  - Computer-aided design tools: simulation, synthesis, …
  - * Hardware design looks like software design
- Emergence of rapid implementation circuit technology
  - Programmable logic devices (PLDs)