§ 1 Review of Digital Logic Design Fundamentals

Logic circuits:

1. Combinational Logic: No memory, present output depends only on the present input
2. Sequential Logic: Has memory, present output depends not only on the present input, but also on the past sequence of inputs

§ 1.1 Combinational Logic

1. Positive logic

   High voltage: Logic 1, Low voltage: Logic 0
2. Devices: gates, adders, MUX, decoder

3. Minimization techniques
   a. Boolean algebra
   b. K-map approach
   c. Tabular approach (Quine-Mclusky Method)
   d. Software (e.g., espresso)
4. Behavior can be specified by a truth table that gives the network outputs for each combination of input values

   Example: one-bit full adder

Using minterms, m-notation / decimal notation

Sum =

Cout =

Using maxterms, M-notation

Sum =

Cout =

§ 1.2 Boolean algebra and Algebraic Simplification

1. Laws and theorems
2. Four ways of simplifying a logic expression
   a. Combining terms

      e.g.: ABC’D’ + ABCD’

   b. Eliminating terms

      e.g.: A’B + A’BC

   c. Eliminating literals

      e.g.: A’B + A’B’C’D’ + ABCD’

   d. Adding redundant terms

      e.g.: WX + XY + X’Z’ + WY’Z’

§ 1.3 Karnaugh Maps

Convenient to simplify logic functions of 3-5 variables

1 indicates that minterm is present

0 indicates that minterm is absent

x indicates don’t care

The variable values along the edge of the map are ordered so that adjacent squares on the map differ in only one variable.

e.g.1: e.g.2:

§ 1.4 Design with NAND and NOR gates

In many technologies, implementation of NAND and NOR gates is easier than AND or OR gates

§ 1.5 Hazards in combinational circuits

When the input to a combinational network changes, unwanted switching transients may appear at the output

Reason: Different paths from input to output have different propagation delays

1. Static 1 (0) hazard

   If, in response to an output change and for some combination of propagation delays, a network output may momentarily go to 0 (1) when it should remain a constant 1 (0), we say the network has a static 1 (0) hazard.

2. Dynamic hazard

   If, when the output is suppose to change from 0 to 1 (or 1 to 0), the output may change three or more times, we say the network has a dynamic hazard.

Static hazard example:

How to remove static hazard?

§ 1.6 Sequential logic devices

flip-flop (FFs) and latches

D-FF:	Q(t+1) = Q(t)
JK-FF:	Q(t+1) = J Q'(t) + K' Q(t)
T-FF:	Q(t+1) = Q(t) xor T

§ 1.7 Mealy machine and Moore machine

Two basic types of sequential networks:

Mealy network: Outputs depend on both present state and present inputs

➔ General model: combinational network is used to generate outputs and next state

Moore network: Outputs depend only on the present state

§ 1.8 Digital Systems Design

1. Examples: Computer, data acquisition system, digital control, communications systems, appliances, e.g., car, microwave oven, washer, drier, etc.
2. Digital systems consists of Hardware + [Software]

   Software exist if a microprocessor is included ➔ more complex, flexible
3. Two parts of a digital hardware

Datapath	+	Controller
Processing unit (work unit)		Control unit
Consists of working components		(management unit)
(e.g., address, memory, decoders, counters, registers, mux, ……)		
Usually repeated and voluminous		

4. *For datapath design*: Strive for efficiency, that is, low hardware, higher speed, minimize # of FFs and # of gates
5. *For controller design*: Unique to each system, usually many I/O but simple circuit Clarity, flexible are important

   ➔ Use design techniques that may not lead to an efficient design but facilitate ease of design and modification

   ➔ One FF per state (one-hot design)
6. Two types of digital system circuits

   Combinational circuit: memoryless

   Sequential circuit: has memory

   ➔ Synchronous sequential circuit: circuits has explicit memory elements such as FFs

   ➔ Asynchronous sequential circuit: w/o FFs, may or may not have latches, very complex and rarely used, not discussed in course
7. Digital circuit design
   - Combinational parts:
     - Logic circuits where outputs depend only on the current input combination
     - gates, decoders, adders, MUXs, etc.
     - Minimization techniques

   - Sequential parts: consist nothing but FFs
     - Circuit where outputs at any time not only depends on its current input, but also the past sequence of inputs that have been applied
       - Circuit with memory
     - Latch, FF, registers, shift registers, counters

- Design procedure
  - Two standard ways to present an output
    - Choose one FF for each state (one-hot design)
    - Use # of FFs ≥ $\log_2$(# of states), design procedure – ASM chart

8. The output of the combinational circuits will decode the input of the sequential circuit, the inputs can be from the output of the sequential circuits (feedback) or from the external input.

   Always, don’t let the primary outputs hang around combinational logic
   ➔ Synthesis tools work best when outputs registered

9. Speed of a digital circuit ➔
   1. Propagation delay of the combinational circuits

      Consider the worst case, i.e., slowest, signal that comes out last
   2. Clock speed