課程名稱:

數位系統設計

授課教授: 台大網路與多媒體研究所 歐陽明 教授

課程編號: 902 365000

國立台灣大學資訊工程學系課程內容綱要

課程名稱 中文：數位系統設計 英文：		課程編號 902 36500
必修或選修：必	學分：3	修習年級：三
每週時數 講演：3 小時 實習或實驗： 小時		預修科目： 同修科目：

內容綱要：（中、英文）

1. Introduction to Boolean Algebra
2. The Process of Design , Rapid Electronic System Prototyping
3. Minimization of Boolean Function
4. Combinational Circuits
5. Programmable and Steering Logic (PLA , PAL , Gate Array , Multiplexers , etc)
6. Sequential Logic Design
7. Finite State Machine Design (VHDL , ABEL Languages , ASM Chart , etc)
8. Case Study : ALU Design , Memory Control

教科書：

Contemporary Logic Design, by Randy H. Katz, 1994 (台北圖書)

主要參考書：

Introduction to Digital Logic Design, John P. Hays, Addison Wesley, 1993 (開發)

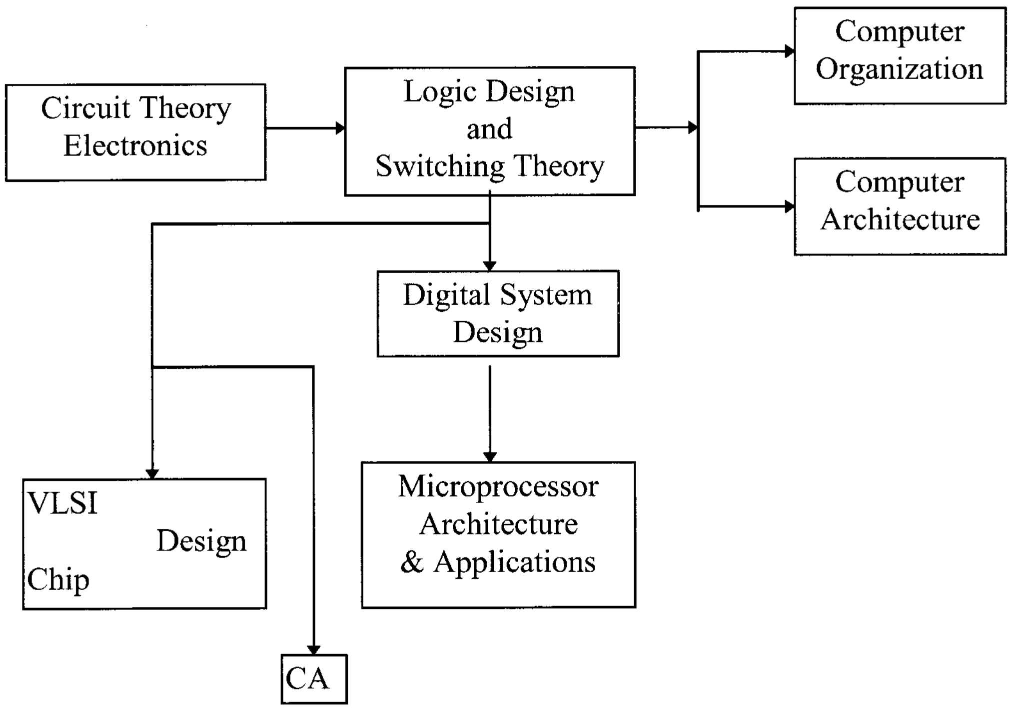
備 註：

歐陽明

台大網路與多媒體研究所教授

歐陽明教授

Logic Design and Switching Theory



Logic Design is

the most fundamental and the most important hardware

course in Computer < Science / Engineering

Logic Design (hardware terminology) → CE

\+ → CSIE

Switching Theory (Algebraic terminology) → CS

How to < learn / study this course??

(i) practising d practising

(ii) taking "good design examples" as references

(iii) complete the design projects

Contents:

Grade:

1. Homeworks : 33 % (6～7 hws)
2. Midterm : 33 %
3. Final : 34 %
4. Design project (presentation) : Extra %

1994 FALL : Contemporary Logic Design, by Randy H. Katz, Benjamin Cummings, 台北圖書公司 U.C.Berkeley.

Text Books : (before 1993)

1. Introduction to Logic Design, by Sajjan G. Shiva, 1988
   (開發) TK 7868, D5, S433

2. Modern Logic Design, by Richard S. Sandige, 1990
   TK 7874, 1990 11613874

References : class notes.
since Fall, 1993

new : Intro. to Digital Logic Design John Hayes,
Addison Wesley, 開發

Testing for Backgrounds:

1. $(.345)_{10} = (?)_2 = (?)_8$

2. Use a 5-bit 2's complement representation (with one sign bit and 4 magnitude bits) to perform the following operations:
   (i) 4-5 (ii) -5-4

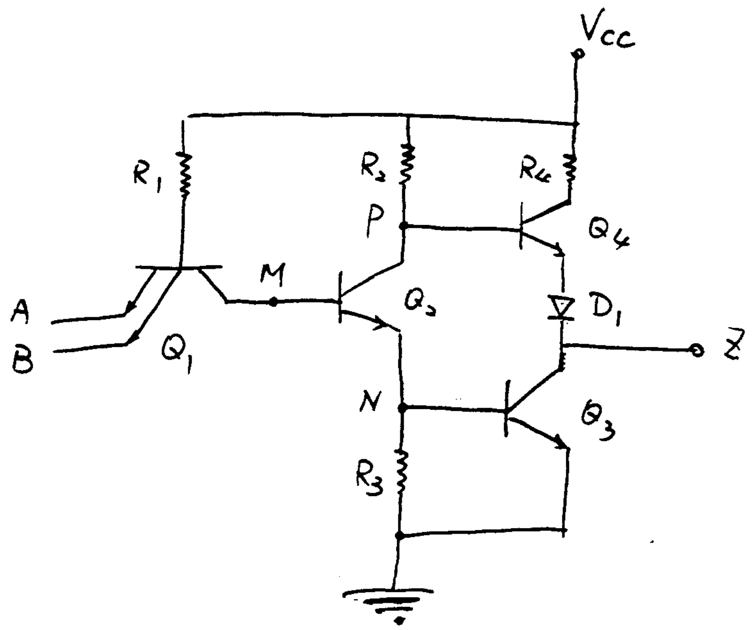
3. Prove the following equalities:
   (i) AB + A'C = (A+C)(A'+B)
   (ii) (QR'+P'R')' = R + PQ'

4. Simplify/Minimize the following:
   (i) F(X,Y,Z) = Σm(1,2,3,6,7)
   (ii) F(X,Y,Z) = Σm(0,1,2,7,8,9,10)
   (iii) F(W,X,Y,Z) = Σm(0,2,4,6,8) + d(10,11,12,13,14,15)

5. Transfer the following AND-OR circuits into NAND-NAND Circuits:

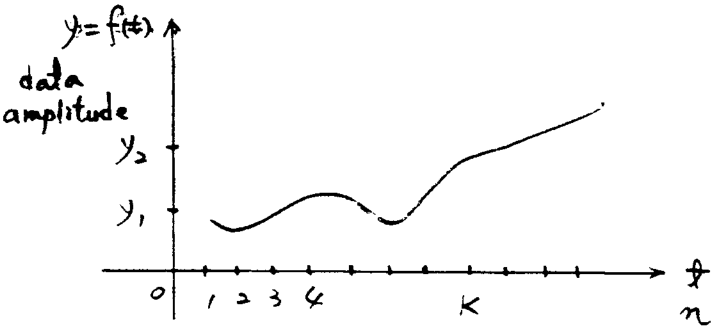
6. Design a circuit (comparator) to compare two 4-bit numbers A and B,
the output $Z = 0$ if $A = B$
$Z = 1$ if $A \neq B$.

7. Complete the following voltage Table.
(TTL NAND with totem pole outlet)



A	B	Q_1	M	Q_2	N	P	Q_3	Q_4	Z
0	0	on	0.4	?	0.2	5	?	on	?
0	5	on	0.4	off	?	?	?	on	?
5	0	on	0.4	off	0.2	5	off	?	?
5	5	on	1.4	?	?	?	on	off	?

Chapter I Number Systems and Codes



t / y	continuous-valued	discrete-valued
continuous-valued	Analog	discrete-time
discrete-valued	Sample-data	digital

Advantages of digital processing:

- accuracy: noise immurity, signal reshapping
- reliability: insensitive to environment
- IC technology ↗
- Easy to program (programmability)

.

.

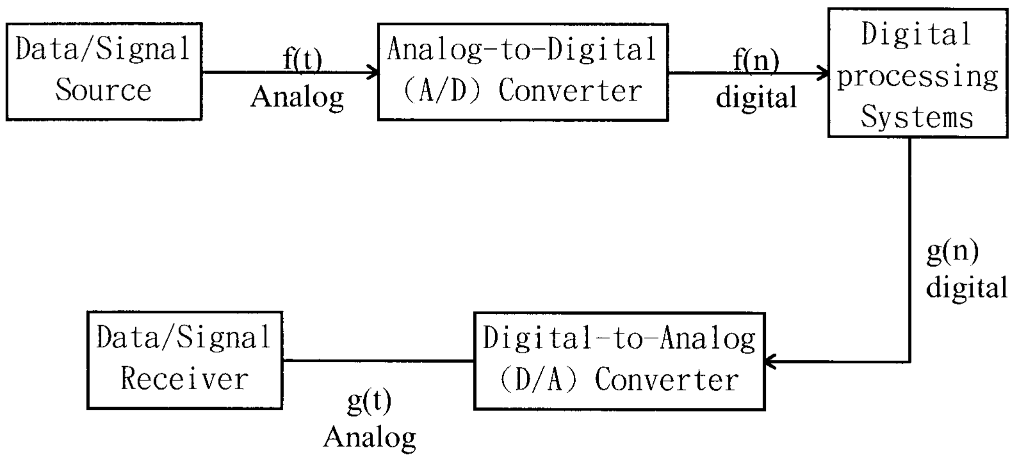
.

Disadvantages as compared with the analog system

.

.

Speed



Data/Signal Source
f(t)
Analog
Analog-to-Digital (A/D) Converter
f(n)
digital
Digital processing Systems
g(n)
digital
Digital-to-Analog (D/A) Converter
g(t)
Analog
Data/Signal Receiver

Chap 1 Introduction

- Combinational circuits — Memory: No
- Sequential circuits — Memory: Yes

    finite-state machine (state M)

    for synchronization: clock

- Design Goals

    Correctness

    Cost $C_T = C_D/N + C_M + C_S$ (Sec 1.6)

    Performance

    delay time: propagation delay

    memory: access time

    communication: bandwidth

    CPU: MIPs

    and Compatibility, Power Consumption,

    Reliability, Testability

    ↑

    design for testability

example:(p31) Graphics Board give your explanations.

eg. budget = US$500

refresh your memory

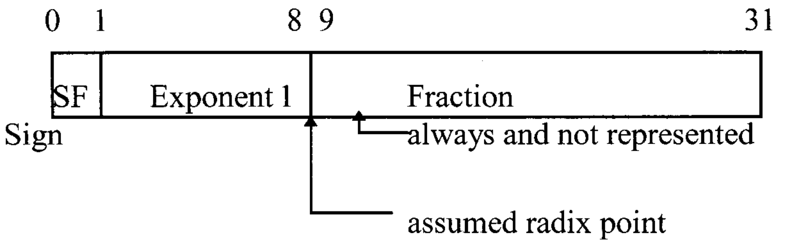
(1) Binary numbers

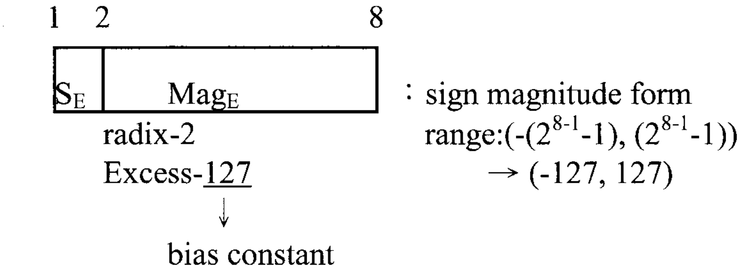
(2) Floating Point

(3) Error Control coding (new)

(4) Logic levels & time delay

IEEE Standard floating-Point Representation





- The MSB of the fraction in the normalized floating-point form is always 1. This 1 is not specifically represented in the IEEE representation but is assumed to exist. Thus an extra bit of accuracy is obtained, since a 24-bit fraction can be represented in only 23 (31-9+1) bits.

- The exponent field is in sign magnitude form.

Ex: Represent (10100.10011) in the floating-point form:

$(10100.10011) = (.1011001001 1) \times 2^5$

SF = 0
exponent = 5 + 127 = 132 = (110000100)
fraction = 010010011

↑ the most significant 1 not shown

=>

0,10000100,0100100110...0

remark:

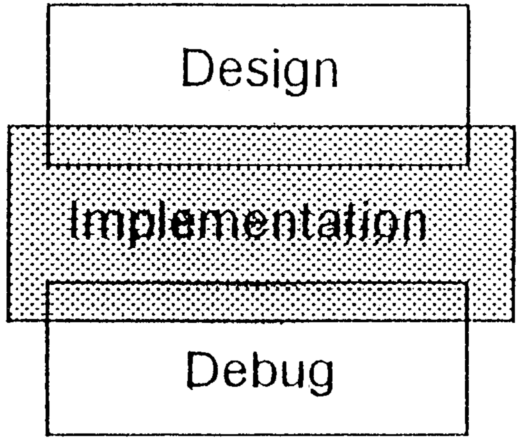
This representation allows only 24 bits for fraction. If the numbers must be represented with more accuracy, the representation extends to the 2nd word containing the remaining fraction bits.

(Double Precision Representation)

↑ not increasing of the dynamic range

A12

The Process Of Design (Chapter 1) Sec 1.1



Design

Initial concept: what is the function performed by the object?
Constraints: How fast? How much area? How much cost?
Refine abstract functional blocks into more concrete realizations

Implementation

Assemble primitives into more complex building blocks
Composition via wiring
Choose among alternatives to improve the design

Debug

Faulty systems: design flaws, composition flaws, component flaws
Design to make debugging easier
Hypothesis formation and troubleshooting skills

The Art Of Design: Refinement of Representations

1. Functional Specification/What the System Does

   Ex: Traffic Light Controller

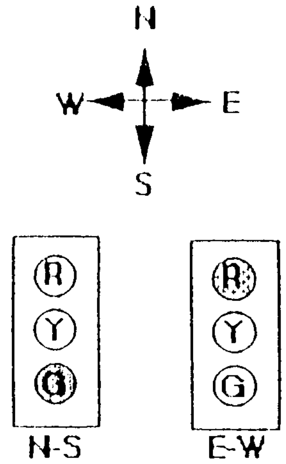
   Lights point in the directions N, S, E, W

   Illuminates the same lights N as S and E as W

   Cycles thru the sequence GREEN-YELLOW-RED

   N-S and E-W never GREEN or YELLOW at the same time

   Stay GREEN for 45 seconds, yellow for 15, red for 60



2. Performance Constraints/Requirements to be Met

   speed: compute changes in under 100 ms

   power: consume less than 20 watts

   area: implementation in less than 20 square cm

   cost: less than $20 in manufacturing costs

Motivation

Dramatic Change in the Way Industry Does Hardware Design

Pervasive use of Computer-Aided Design Tools

Deemphasis on hand design methods

Emphasis on abstract design representations

Hardware design begins to look like software design

Emergence of Rapid Implementation Circuit Technology

Programmable rather than discrete logic

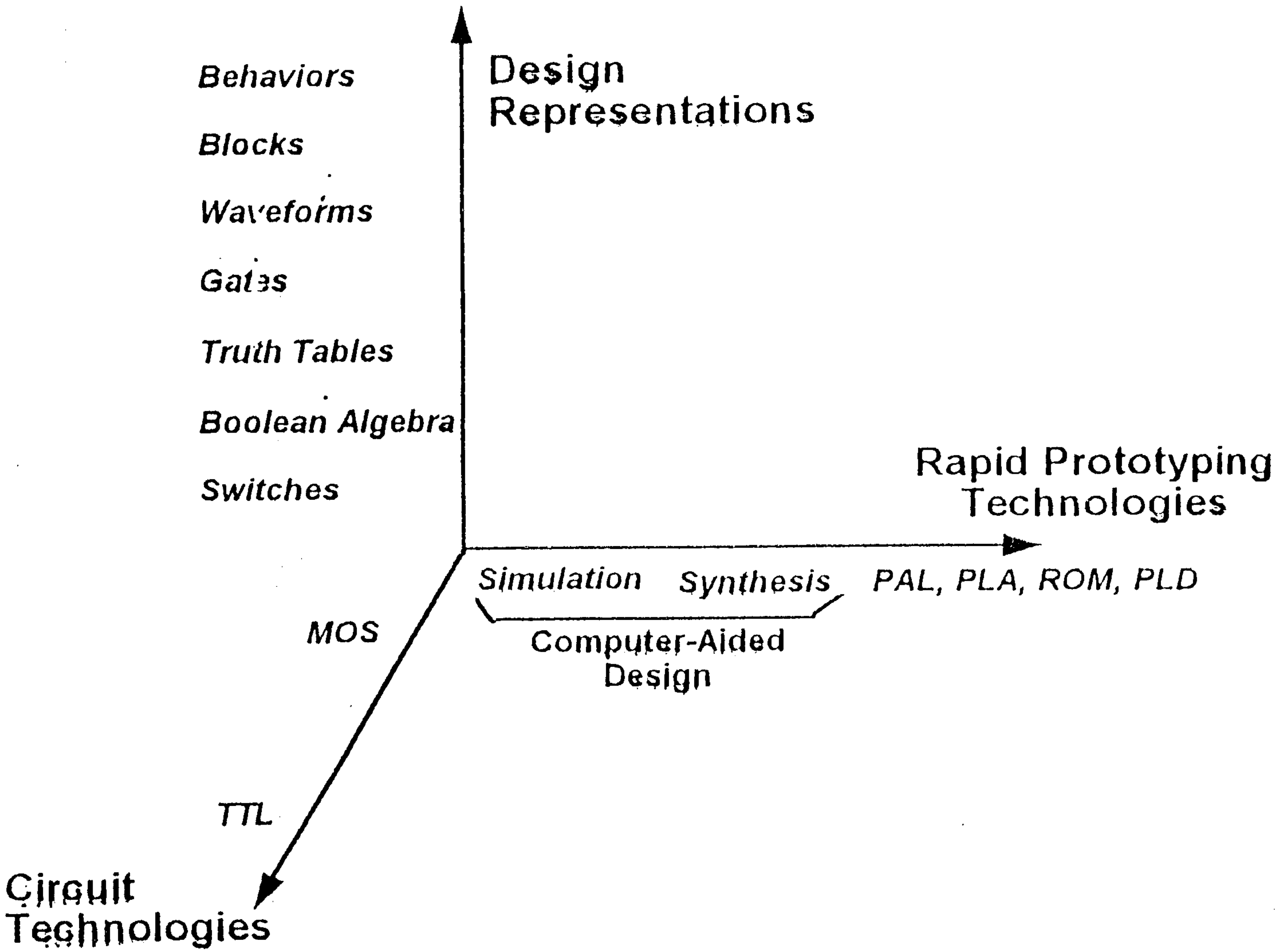
Importance of Sound Design Methodologies

Synchronous Designs

Rules of Composition

The Elements of Modern Design

Representations, Circuit Technologies, Rapid Prototyping



A23

Digital Hardware Systems

Sequential logic

inputs and outputs overlap
outputs depend on inputs *and* the entire history of execution!

network typically has only a limited number of unique configurations
these are called *states*
e.g., traffic light controller sequences infinitely through four states

new component in sequential logic networks:
storage elements to remember the current state

output and new state is a function of the inputs and the old state
i.e., the fed back inputs are the state!

Synchronous systems

period reference signal, the clock, causes the storage elements to accept new values and to change state

Asynchronous systems

no single indication of when to change state

The Art of Design: "To Design Is To Represent"

1. *English language specification*

   easy to write, but not precise and subject to ambiguity

2. *Functional description*

   more precise specification

   flow charts, program fragments

3. *Structural description*

   complex components decomposed into compositions of less complex components

4. *Physical description*

   the design in terms of most primitive building blocks, e. g., logic gates or transistors