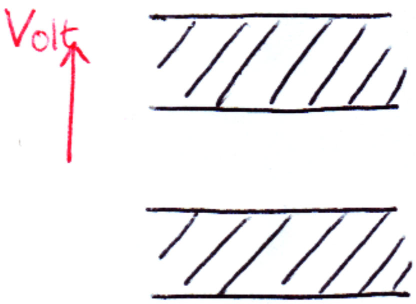
P3360/AEP3630

Lecture 23

Analog signals info represented by continuously varying I & V

- sensitive to drifts, noise, etc.
- 1 part in 10^6 accuracy is difficult

Digital signals



} V_{HI} = "1" (e.g. (3-5V))

} forbidden range

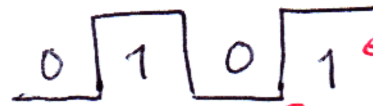
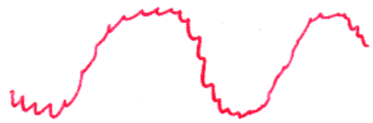
} V_{LO} = "0" (0-2V)

Positive logic: $V_{HI} > V_{LO}$

"1" "0"

negative logic

- only two discrete voltage ranges "0" and "1" FALSE and TRUE
- the info typically is encoded
- digital circuits perform logical op ("computer") (on "0" and "1"'s)
- digital signals can be recovered exactly
- e.g. 1 part in 10^{13} accuracy is easy

	analog	digital
original signal (vs. time)		 digital repr. of signal
signal + noise		
recovered ?	No	 Yes (e.g. Schmitt trigger)
arbitrary shapes ?	 yes	Must be encoded, i.e. converted into the sequence of "0" and "1"'s

Digital representation of info

e.g. temperature 43.124... °F

(1) <u>truncate</u> to some precision, e.g. 1°F

$43_{(10)} = 4\times10^1 + 3\times10^0$ ← decimal representation

(10) ↖ decimal base

(2) need <u>binary</u> representation. Several options:

a) binary numbers

$43_{10} = 1\times2^5 + 0\times2^4 + 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0$

$43_{10} = 101011_{(2)}$ ← binary base

↑↑↑ bits

- each bit = binary-valued variable
- byte = 8 bits

b) use binary codes (lookup tables)

Example 1. Binary-coded decimal

$3360_{10} \rightarrow 0011\ 0011\ 0110\ 0000_{BCD}$

each digit 0-9 is represented by 4-bit binary

Example 2. Gray code - used to represent successive numbers

decimal	binary (3-bit)		Gray code
0	000		000
1	001		001
2	010		011
3	011	} glitch	010
4	100		110
5	101		111
6	110		101
7	111	} possible	100
	1000		

only 1 bit changes on successive number increments

pro: avoids "glitches" when multiple bits need to change simultaneously

Example 3. ASCII - american standard code for info inter-change

"a" = 1100001 "A" = 1000001

"b" = 1100010 "B" = 1000010

"c" = 1100011 "C" = 1000011

7-bit to represent "printable" characters

$75_{10} \rightarrow \_\_\__2$?

64 - 32 - 16 - 8 - 4 - 2 - 1

1 (11) 0 0 1 (3) 0 1 (1) 1

$\Rightarrow 1001011_2$

binary to gray? $101101_2 \rightarrow \_\_\__{gray}$? ($45_{10}$)

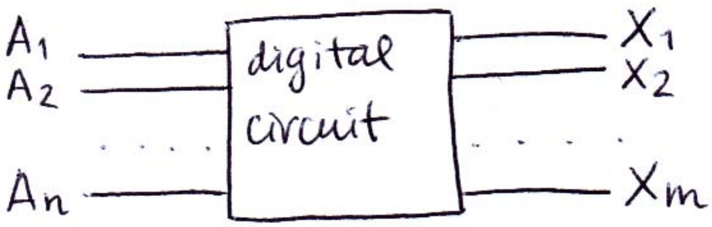
MSB 10 01 11 10 01

↘ 1 1 1 0 1 1

$\Rightarrow 111011_{gray}$

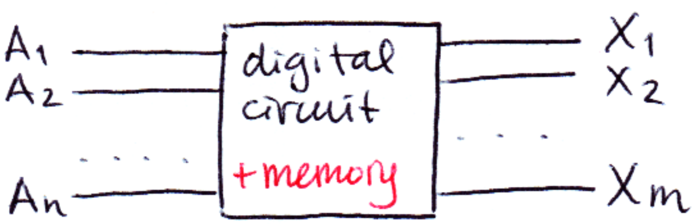
Types of digital circuits

In general, a digital circuit has n inputs A_i, $i = 1 \ldots n$
m outputs X_j, $j = 1 \ldots m$



– If output depends only on present inputs

$$X_j = f_j(A_1, A_2, \ldots, A_n) \Rightarrow \text{"combinational logic"}$$



– If output depends both on present and past inputs (thru memory)

$$X_j = f_j(A_1 \ldots A_n;\ \underbrace{M_1 \ldots M_\ell}_{\text{memory (flip-flops)}}) \Rightarrow \text{"sequential logic"}$$

Truth table (lookup table) – list all input combinations and corresponding outputs ("0" and "1"'s only)

size: $2^{\overbrace{n}^{\text{inputs}}} \times \underbrace{m}_{\text{outputs}}$

e.g. 2 inputs + 1 output : $2^2 = 4$
5 inputs + 4 outputs : $2^5 \times 4 = 128$
16 inputs + 8 outputs : $8 \times 2^{16} = 524k$