CS210201 Logic Design 數位邏輯設計

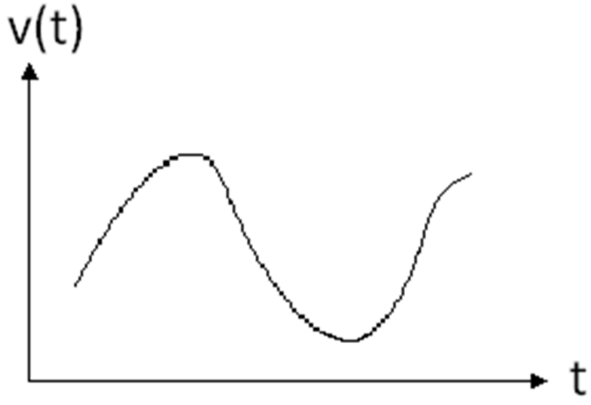
Unit 1

Introduction / Number Systems and Conversion

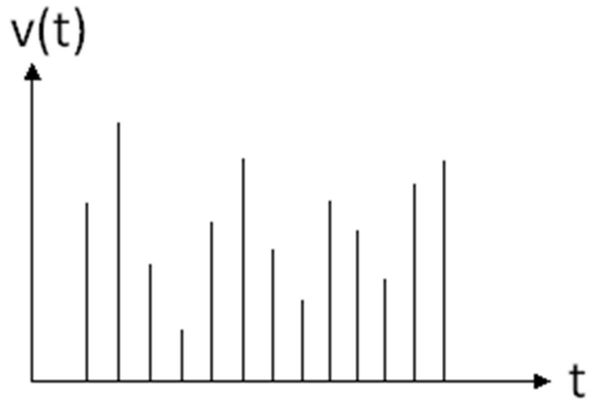
Outline

- Digital systems and switching circuits
- Number systems and conversion
- Binary arithmetic
- Negative numbers
- Binary codes

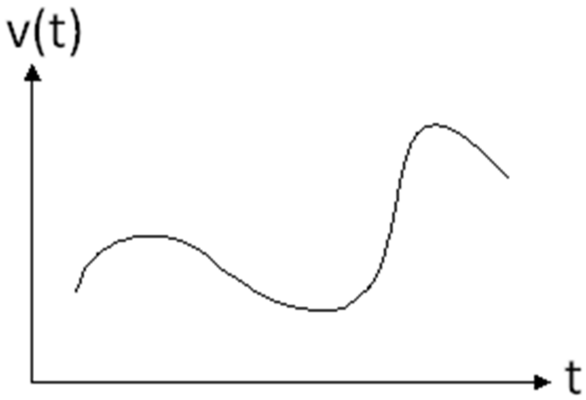
Digital Systems and Switching Circuits (1/3)



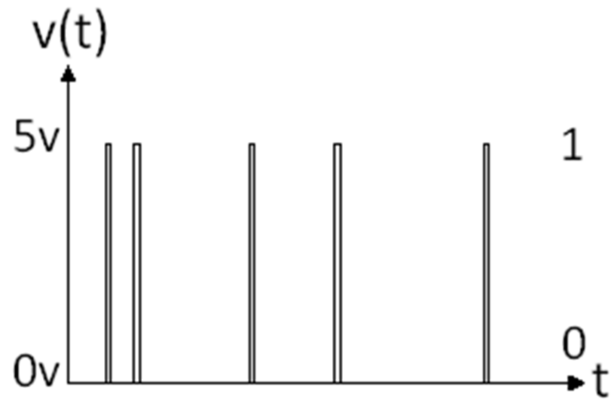
Continuous signal



Discrete signal

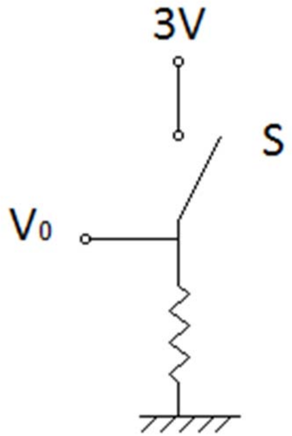


Analog signal



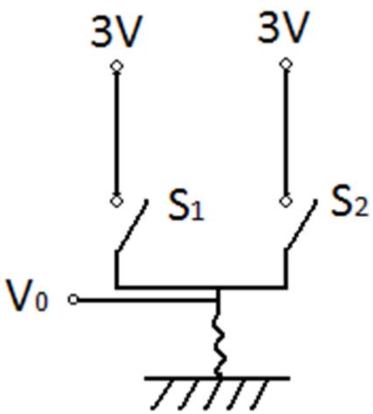
Digital signal

Digital Systems and Switching Circuits (2/3)

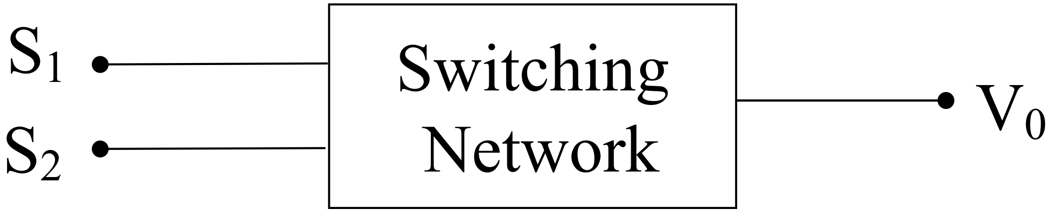


S ON "1" (3V)
OFF "0" (0V)

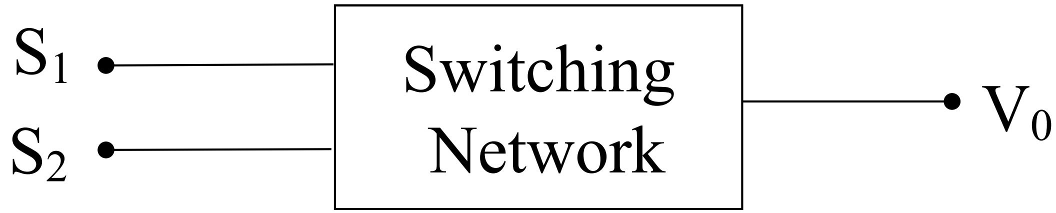
Switching circuit



S_1 OR S_2 ON (3V) "1"
S_1 AND S_2 OFF (0V) "0"



Digital Systems and Switching Circuits (3/3)



- Combinational network
  - V_0 : function of S_1, S_2 present values
- Sequential network
  - V_0 : function of S_1, S_2 both present & previous values
  - memory behavior
- Switches
  - realized by transistors
  - transistor level, gate level, module level (adder, arithmetic unit….)

Number Systems & Conversion (1/9)

- Number Systems
  - Base 10：953.78_{10} (Decimal) $= 9\times10^{2} + 5\times10^{1} + 3\times10^{0} + 7\times10^{-1} + 8\times10^{-2}$
  - Base 2：1011.11_{2} (Binary) $= 1\times2^{3} + 0\times2^{2} + 1\times2^{1} + 1\times2^{0} + 1\times2^{-1} + 1\times2^{-2}$

$$= 8 + 0 + 2 + 1 + \frac{1}{2} + \frac{1}{4} = 11\frac{3}{4} = 11.75_{10}$$

Number Systems & Conversion (2/9)

- Base (Radix) R : (0, 1, ⋯, R-1)   Base 4 : (0, 1, 2, 3)

- $N = (a_4 a_3 a_2 a_1 a_0 . a_{-1} a_{-2} a_{-3})_R$

$$= a_4 \times R^4 + a_3 \times R^3 + a_2 \times R^2 + a_1 \times R^1 + a_0 \times R^0$$

$$+ a_{-1} \times R^{-1} + a_{-2} \times R^{-2} + a_{-3} \times R^{-3} \quad , \quad 0 \le a_i \le R-1$$

- Octal $147.3_8 = 1 \times 8^2 + 4 \times 8^1 + 7 \times 8^0 + 3 \times 8^{-1} = 64 + 32 + 7 + \frac{3}{8}$

$$= 103.375_{10}$$

- Hexadecimal Base16:(0, 1, 2, …, 9, A, B, C, D, E, F)

- $A2F_{16} = 10 \times 16^2 + 2 \times 16^1 + 15 \times 16^0 = 2560 + 32 + 15$

$$= 2607_{10}$$

Number Systems & Conversion (3/9)

- Conversion

Decimal $\Rightarrow$ R

$53_{10} \Rightarrow$ Binary

$$\begin{array}{cccccc} a_5 & a_4 & a_3 & a_2 & a_1 & a_0 \end{array}$$

$$53_{10} = 1\ 1\ 0\ 1\ 0\ 1_2$$

$$\begin{array}{lll}
2\,)\ 53 & & \\
2)\ 26 \ldots\ldots. & 1 & a_0 \\
2)\ 13 \ldots\ldots. & 0 & a_1 \\
2)\ 6 \ \ldots\ldots. & 1 & a_2 \\
2)\ 3 \ \ldots\ldots. & 0 & a_3 \\
2)\ 1 \ \ldots\ldots. & 1 & a_4 \\
\quad 0 \ \ldots\ldots. & 1 & a_5
\end{array}$$

Number Systems & Conversion (4/9)

- $N_{10} = (a_n a_{n-1} \ldots a_2 a_1 a_0)_R = a_n R^n + a_{n-1} R^{n-1} + \ldots\ldots + a_2 R^2 + a_1 R^1 + a_0$

  $= R(a_n R^{n-1} + a_{n-1} R^{n-2} + \ldots\ldots + a_2 R^1 + a_1) + a_0$

- $\dfrac{N_{10}}{R} = a_n R^{n-1} + a_{n-1} R^{n-2} + \ldots\ldots + a_2 R + a_1$ ------------- Remainder a_0

  $= R(a_n R^{n-2} + a_{n-1} R^{n-3} + \ldots\ldots a_3 R + a_2) + a_1$

- $\dfrac{N_{10}}{R^2} = a_n R^{n-2} + \ldots\ldots a_3 R + a_2$ ------------------------- Remainder a_1

- $\dfrac{N_{10}}{R^3} = a_n R^{n-3} + \ldots\ldots + a_3$ -------------------------------- Remainder a_2

⋮

⋮

Number Systems & Conversion (5/9)

$$F_{10} = (.a_{-1}\, a_{-2} \ldots . a_{-m}\,)_R = a_{-1}\, R^{-1} + a_{-2}R^{-2} + \ldots\ldots + a_{-m}\, R^{-m}$$

$$FR = a_{-1} + a_{-2}\, R^{-1} + \ldots\ldots + a_{-m}\, R^{-m+1} = a_{-1} + F_1$$

$$F_1 R = a_{-2} + a_{-3}\, R^{-1} + \ldots\ldots + a_{-m}\, R^{-m+2} = a_{-2} + F_2$$

$$F_2 R = a_{-3} + F_3$$

⋮

⋮

Number Systems & Conversion (6/9)

$$Ex : .625_{10} \Rightarrow .(\qquad\qquad)_2$$

$$F = .625 \qquad F_1 = 250 \qquad F_2 = .50$$

$$\times\ 2 \qquad \times\ 2 \qquad \times\ 2$$

$$1.250 \qquad 0.500 \qquad 1.000$$

$$\uparrow \qquad \uparrow \qquad \uparrow$$

$$a_{-1} \qquad a_{-2} \qquad a_{-3}$$

$$\Rightarrow .101_2$$

Number Systems & Conversion (7/9)

$$0.7_{10} = (0.1\overline{0110})_2$$

$$
\begin{array}{lr}
 & .7 \\
 & \times\ 2 \\
\hline
a_{-1} \rightarrow & (1).4 \\
 & \times\ 2 \\
\hline
a_{-2} \rightarrow & (0).8 \\
 & \times\ 2 \\
\hline
a_{-3} \rightarrow & 1.6 \\
 & \times\ 2 \\
\hline
a_{-4} \rightarrow & 1.2 \\
 & \times\ 2 \\
\hline
a_{-5} \rightarrow & 0.4
\end{array}
$$

Number Systems & Conversion (8/9)

$$Ex.\quad 231.3_4 \Rightarrow (\,63.5151\,)_7$$

$$231.3_4 = 2\times 4^2 + 3\times 4^1 + 1\times 4^0 + 3\times \tfrac{1}{4} = 45.75_{10}$$

$$\begin{array}{l l l}
7\,)\;45 & & \\
\hline
7\,)\;\;\,6 & \cdots\cdots 3 \\
\hline
\quad\;\;\,0 & \cdots\cdots 6
\end{array}$$

$$\begin{array}{r l}
 & .75 \\
 & \times\;\;7 \\
\hline
(5) & .25 \\
 & \times\;\;7 \\
\hline
(1) & .75 \\
 & \times\;\;7 \\
\hline
(5) & .25 \\
 & \times\;\;7 \\
\hline
(1) & .75
\end{array}$$

$$45.75_{10} = (63.5151)_7$$

Number Systems & Conversion (9/9)

Binary to Octal

$$\underbrace{1}\ \overleftrightarrow{101}\ \overleftrightarrow{011}\ \overleftrightarrow{101}\ \overleftrightarrow{110}\ .\ \overleftrightarrow{001}\ \overleftrightarrow{100}_2 = 15356.14_8$$

$$1\quad 5\quad 3\quad 5\quad 6\quad .\quad 1\quad 4\ _8$$

Binary to Hexadecimal

$$1\ \overleftrightarrow{1010}\ \overleftrightarrow{1110}\ \overleftrightarrow{1110}\ .\ \overleftrightarrow{0011}_2 = 1AEE.3_{16}$$

$$1\quad A\quad E\quad E\quad .\quad 3$$

Binary Arithmetic (1/2)

Addition

$$
\begin{array}{rcccccc}
 & & & 1 & 1 & 1 & \\
 & 13_{10} & = & 1 & 1 & 0 & 1 \\
+ & 11_{10} & = & 1 & 0 & 1 & 1 \\
\hline
 & & 1 & 1 & 0 & 0 & 0 \quad = 24_{10}
\end{array}
$$

Subtraction

$$
\begin{array}{cccccc}
 & 1 & 1 & 1 & 0 & 1 \\
- & 1 & 0 & 0 & 1 & 1 \\
\hline
 & 0 & 1 & 0 & 1 & 0
\end{array}
$$

Binary Arithmetic (2/2)

Multiplication

				1	1	0	1	=	13_{10}
×				1	0	1	1	=	11_{10}
				1	1	0	1		
			1	1	0	1			
		0	0	0	0				
	1	1	0	1					
1	0	0	0	1	1	1	1	=	143_{10}

Division

						1	1	0	1
1011	)	1	0	0	0	1	1	1	1
			1	0	1	1			
			0	1	1	0	1		
				1	0	1	1		
					0	1	0	1	1
						1	0	1	1
									0

Negative Numbers (1/10)

+N	Positive integers (for all systems)	-N	Negative integers		
			Sign and magnitude	1's complement $\overline{N}$	2's complement N*
+0	0000	-0	1000	1111	-----
+1	0001	-1	1001	1110	1111
+2	0010	-2	1010	1101	1110
+3	0011	-3	1011	1100	1101
+4	0100	-4	1100	1011	1100
+5	0101	-5	1101	1010	1011
+6	0110	-6	1110	1001	1010
+7	0111	-7	1111	1000	1001
		-8	-----	-----	1000

Negative Numbers (2/10)

- Sign-magnitude
  - (sign bit) + (positive magnitude)
  - Hard to implement in hardware
- 1's complement
  - $\overline{N} = (2^n - 1) - N$
- 2's complement
  - $N^* = 2^n - N$, for n = 4, -N => 16-N
  - e.g., -3 => 16-3 = 13 = 1101_2
- Easy way to form 1's complement
  - Complementing N bit-by-bit
- Easy way to form 2's complement
  - Complementing N bit-by-bit and then adding 1

Negative Numbers (3/10)

2's complement

- Addition of *n-bit signed binary numbers*
  - Any carry from the sign position is ignored

1. Addition of two positive numbers, sum $< 2^{n-1}$

$$\begin{array}{rr} +3 & 0011 \\ +4 & 0100 \\ \hline +7 & 0111 \end{array} \leftarrow \textit{correct answer}$$

2. Addition of two positive numbers, sum $\geq 2^{n-1}$

$$\begin{array}{rr} +5 & 0101 \\ +6 & 0110 \\ \hline +11 & 1011 \end{array} \leftarrow \text{Wrong answer because of overflow}$$

(+11 requires 5 bits including sign)

Negative Numbers (4/10)

3. Addition of positive and negative numbers (negative numbers has greater magnitude)

+5	0101	
–6	1010	
–1	1111	← (correct answer, 0001 ->1110+1=1111)

4. Addition of positive and negative numbers (positive number has greater magnitude)

$$-A+B=A^{*}+B=(2^{n}-A)+B$$

$$=2^{n}+(B-A)\geqq 2^{n}\quad (B>A,\ \text{carry})$$

Throwing away the last carry is equivalent to subtracting 2^n,

So the result is $(B-A)$

–5	1011	
+6	0110	
+1	(1) 0001	← correct answer when carry from the sign bit is ignored (this is not an overflow)

Negative Numbers (5/10)

5. Addition of two negative numbers, $|sum| \leqq 2^{n-1}$

$-A-B = A^* + B^* = (2^n - A) + (2^n - B)$

$= 2^n + 2^n - (A+B) \geqq 2^n \; (A+B \leqq 2^{n-1}$, carry)

Discarding the last carry yields $2^n - (A+B) = (A+B)^*$,

Which is the correct representation of $-(A+B)$

–3	1101
–4	1100
–7	(1)1001

← correct answer when the last carry is ignored (this is not an overflow 0111→1000+1=1001)

6. Addition of two negative numbers, $|sum| > 2^{n-1}$

–5	1011
–6	1010
–11	(1)0101

← wrong answer because of overflow (-11 requires 5 bits including sign) 1011→0100+1=0101→1 0101

Negative Numbers (6/10)

Example: Add -8 and +19 in 2's complement for a word length of n=8

+8=00001000 → -8=11111000

+19=00010011

$$
\begin{array}{rr}
11111000 & (-8) \\
\underline{00010011} & \underline{+19} \\
\not{(1)}\;00001011 = & +11
\end{array}
$$

(discard the last carry)

Negative Numbers (7/10)

1's complement

- Addition of *n-bit signed binary numbers*
  - *Add the last carry (end-around carry) to the n-bit sum in the position furthest to the right*

1. Addition of two positive numbers, sum $< 2^{n-1}$

$$\begin{array}{rr} +3 & 0011 \\ +4 & 0100 \\ \hline +7 & 0111 \end{array}$$ ← correct answer

2. Addition of two positive numbers, sum $\geqq 2^{n-1}$

$$\begin{array}{rr} +5 & 0101 \\ +6 & 0110 \\ \hline +11 & 1011 \end{array}$$ ← wrong answer because of overflow (+11 requires 5 bits including sign)

Negative Numbers (8/10)

3. Addition of positive and negative numbers
   (negative number with greater magnitude)

+5	0101	
−6	1001	
−1	1110	(correct answer)

4. Addition of positive and negative numbers
   (positive number with greater magnitude)

−5	1010	
+6	0110	
	(1) 0000	
	→ 1	(end-around carry)
	0001	(correct answer, no overflow)

$-A+B=\overline{A}+B=(2^n-1-A)+B$

$=2^n+(B-A)-1 \geqq 2^n \ (B>A,\text{ carry})$

The end-around carry is equivalent to subtracting 2^n and adding 1, so the result is $(B-A)$

Negative Numbers (9/10)

5. Addition of two negative numbers, $|sum| < 2^{n-1}$

```
 -3        1100
 -4        1011
 ---       ----
        (1)0111
        └────→1
           ----
           1000
```

$-A-B = \overline{A} + \overline{B} = (2^n - 1 - A) + (2^n - 1 - B)$
$= 2^n + [(2^n - 1 - (A+B)] - 1 \geqq 2^n$ ($A+B < 2^{n-1}$, carry)

Discarding the last carry yields $2^n - 1 - (A+B) = \overline{(A+B)}$,
which is the correct representation of $-(A+B)$

(end-around carry)
(correct answer, no overflow)

6. Addition of two negative numbers, $|sum| \geqq 2^{n-1}$

```
 -5        1010
 -6        1001
 ---       ----
        (1)0011
        └────→1
           ----
           0100
```

(end-around carry)
(wrong answer because of overflow)

Negative Numbers (10/10)

Example: Add -11 and -20 in 1's complement for a word length of n=8

+11=00001011 +20=00010100

-11=11110100 -20=11101011

$$
\begin{array}{rl}
11110100 & (-11) \\
\underline{11101011} & \underline{+(-20)} \\
(1)\,11011111 & \\
\underline{\quad\quad\quad\quad 1} & \text{(end-around carry)} \\
11100000 = -31 &
\end{array}
$$

Binary Codes (1/4)

Binary-Coded-Decimal (BCD)

8-4-2-1 BCD

9	3	7	2	5
←→	←→	←→	←→	←→
1001	0011	0111	0010	0101

Why Binary Codes ?

Binary Codes (2/4)

Decimal Digit	8−4−2−1 Code (BCD)	6−3−1−1 Code	Excess−3 Code	2−out−of−5 Code	Gray Code
0	0000	0000	0011	00011	0000
1	0001	0001	0100	00101	0001
2	0010	0011	0101	00110	0011
3	0011	0100	0110	01001	0010
4	0100	0101	0111	01010	0110
5	0101	0111	1000	01100	1110
6	0110	1000	1001	10001	1010
7	0111	1001	1010	10010	1011
8	1000	1011	1011	10100	1001
9	1001	1100	1100	11000	1000

Binary Codes (3/4)

Weighted Codes

$$\begin{array}{llll} 8 & 4 & 2 & 1 \\ 6 & 3 & 1 & 1 \quad Ex.\ 1011 = 6+1+1 = 8 \end{array}$$

Excess-3 Codes

$$\begin{array}{llll} 8 & 4 & 2 & 1 \quad + \quad 3\,(0011) \text{ to each code} \\ 0 & 0 & 1 & 1 \quad \rightarrow \quad 0110 \end{array}$$

2-out-of-5 Codes

2 out of 5 digits are “1”

Error-correcting codes

Gray Codes

Each successive decimal digit differs by 1 bit

Binary Codes (4/4)

ASCII Code: Character to Binary

0	0011 0000	O	0100 1111	m	0110 1101
1	0011 0001	P	0101 0000	n	0110 1110
2	0011 0010	Q	0101 0001	o	0110 1111
3	0011 0011	R	0101 0010	p	0111 0000
4	0011 0100	S	0101 0011	q	0111 0001
5	0011 0101	T	0101 0100	r	0111 0010
6	0011 0110	U	0101 0101	s	0111 0011
7	0011 0111	V	0101 0110	t	0111 0100
8	0011 1000	W	0101 0111	u	0111 0101
9	0011 1001	X	0101 1000	v	0111 0110
A	0100 0001	Y	0101 1001	w	0111 0111
B	0100 0010	Z	0101 1010	x	0111 1000
C	0100 0011	a	0110 0001	y	0111 1001
D	0100 0100	b	0110 0010	z	0111 1010
E	0100 0101	c	0110 0011	.	0010 1110
F	0100 0110	d	0110 0100	,	0010 0111
G	0100 0111	e	0110 0101	:	0011 1010
H	0100 1000	f	0110 0110	;	0011 1011
I	0100 1001	g	0110 0111	?	0011 1111
J	0100 1010	h	0110 1000	!	0010 0001
K	0100 1011	I	0110 1001	'	0010 1100
L	0100 1100	j	0110 1010	"	0010 0010
M	0100 1101	k	0110 1011	(	0010 1000
N	0100 1110	l	0110 1100	)	0010 1001
				space	0010 0000