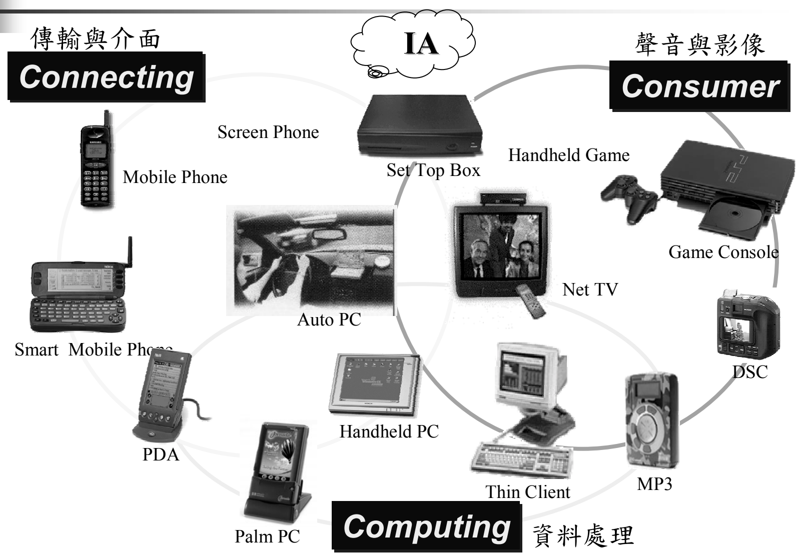
Chapter 1

Binary Systems

Outline

- Introductions
- Number Base Conversions
- Binary Arithmetic
- Binary Codes
- Binary Elements

3C Integration



Milestones for IC Industry

- **1947**: Bardeen, Brattain & Shockley invented the <u>transistor</u>, foundation of the IC industry
- **1958**: Kilby invented integrated circuits (ICs)
- **1968**: <u>Noyce and Moore</u> founded Intel
- **1971**: Intel announced 4-bit 4004 microprocessors (2300 transistors)
- **1976/81**: Apple / IBM PC
- **1985**: Intel began focusing on microprocessors
- Today, Intel-P4 has > 10M transistors
  - up to 1.13 GHz; 0.18 um
- Semiconductor/IC: #1 key field for advancing into Y2K (Business Week, Jan. 1995)

The First Transistor

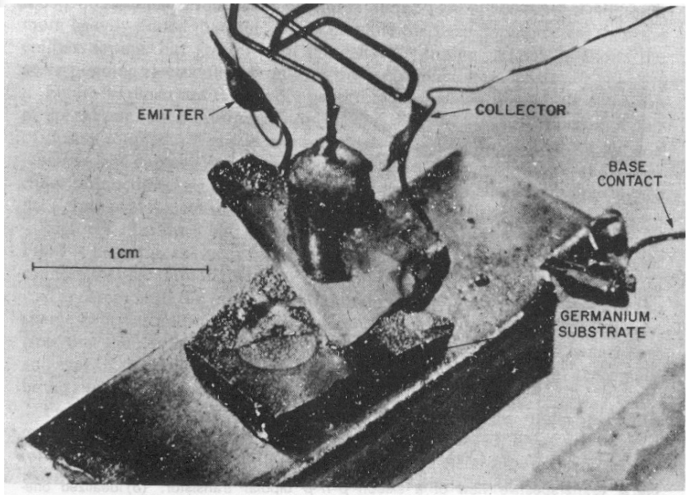
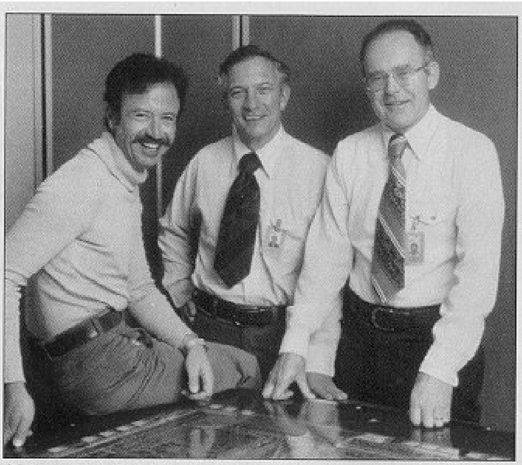


Fig. 1 The first transistor.[1]

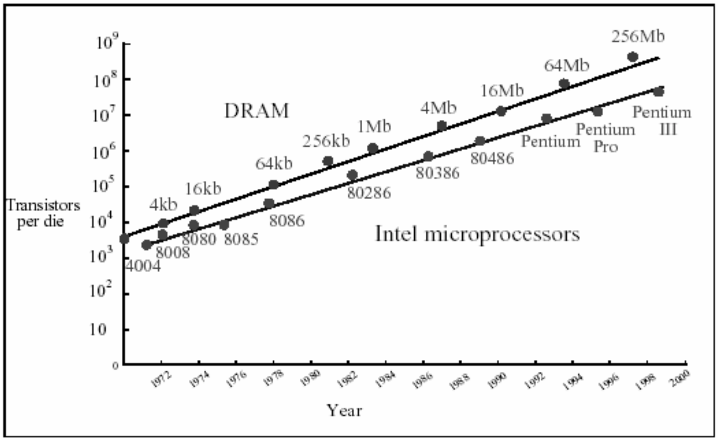
Pioneers of the Electronic Age



[2] Gordon Moore [right] relaxes with fellow pioneers of the electronic age: Robert Noyce [center] and Andrew Grove [left]. Moore and Noyce contributed to the development of the planar IC. Grove is now president and chief executive officer of Intel Corp.

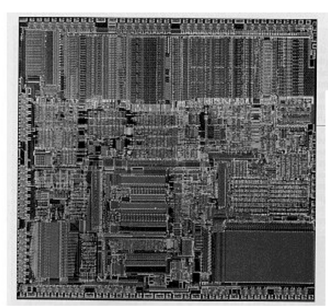
Moore's Law

- Logic capacity doubles per IC per year at regular intervals (1965)
- Logic capacity doubles per IC every 18 months (1975)



The Dies of Intel CPUs

Pentium Pro



386

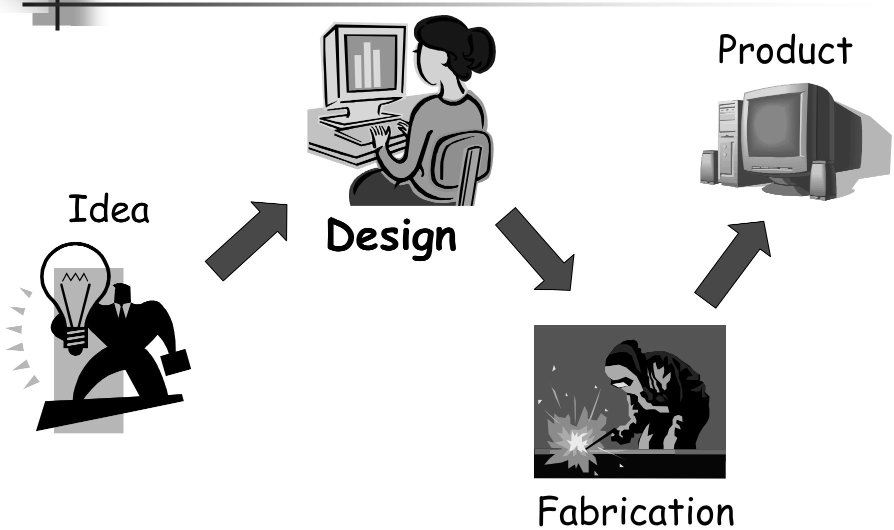
4004

Semiconductor Technology Roadmap

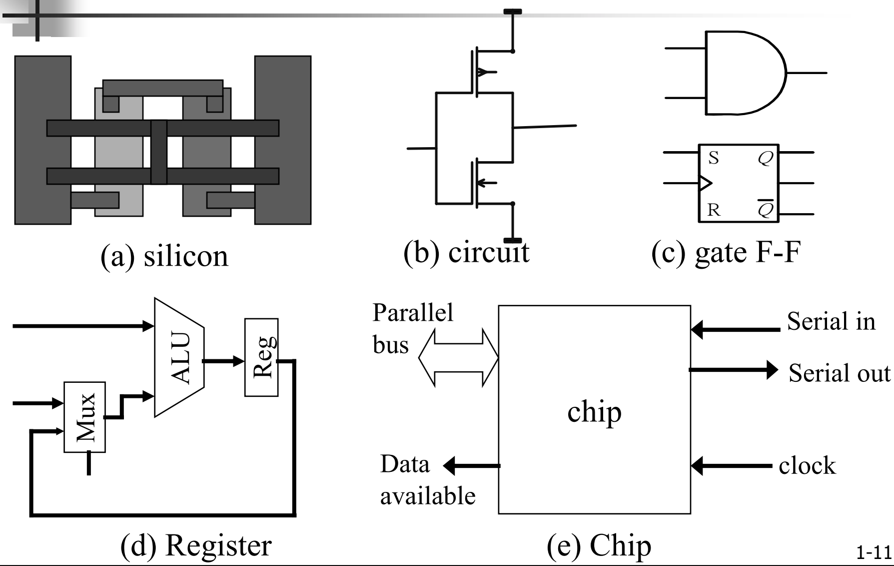
Year	1997	1999	2002	2005	2008	2011	2014
Technology node (nm)	250	180	130	100	70	50	35
On-chip local clock (GHz)	0.75	1.25	2.1	3.5	6.0	10	16.9
Microprocessor chip size (mm^2)	300	340	430	520	620	750	901
Microprocessor transistor/chip	11M	21M	76M	200M	520M	1.40B	3.62B
Microprocessor cost/transistor ($x10^{-8}$ USD)	3000	1735	580	255	110	49	22
DRAM bits per chip	256M	1G	4G	16G	64G	256G	1T
Wiring level	6	6-7	7	7-8	8-9	9	10
Supply voltage (V)	1.8-2.5	1.5-1.8	1.2-1.5	0.9-1.2	0.6-0.9	0.5-0.6	0.37-0.42
Power (W)	70	90	130	160	170	175	183

Source: SIA99

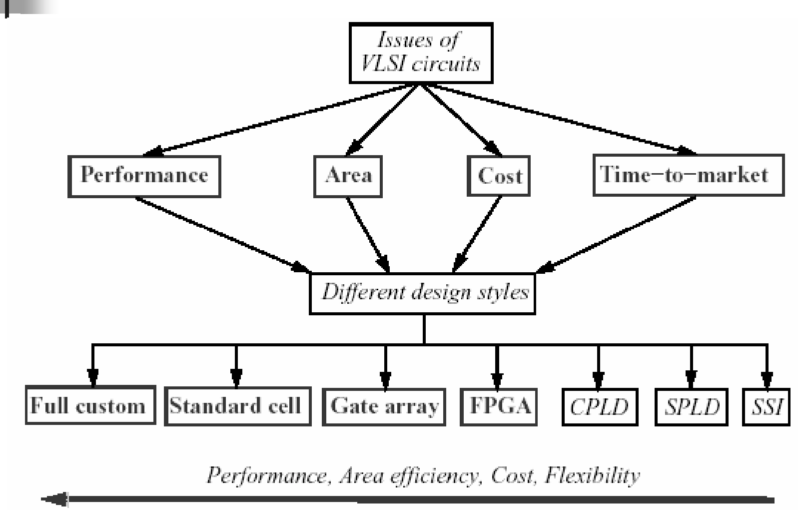
Product Creation



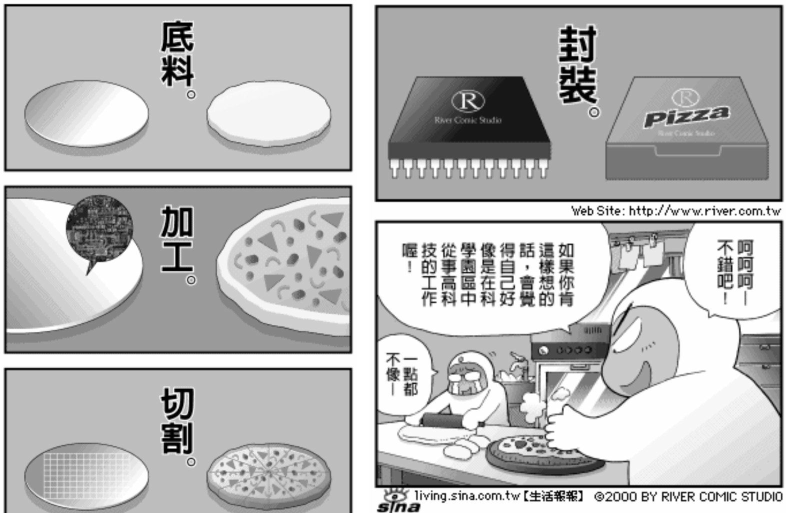
Design on Different levels



Design Styles

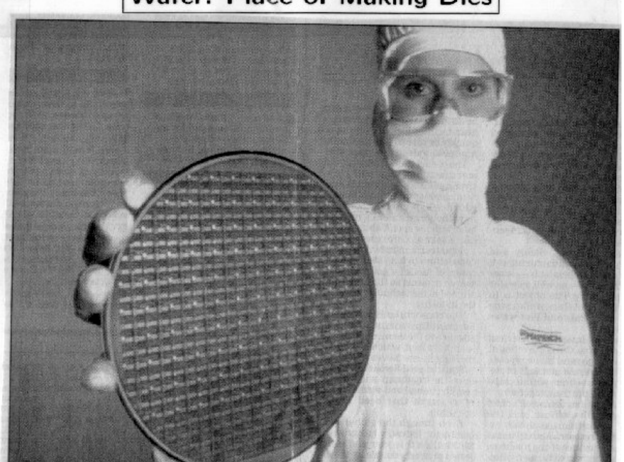


IC Fabrication



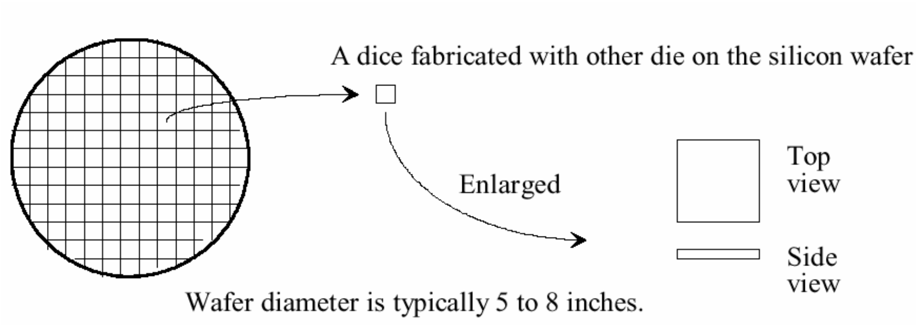
Silicon Wafers

Wafer: Place of Making Dies



A coalition of semiconductor makers, including Austin's Sematech, has released a report detailing where the industry is headed. Among the projects is increasing silicon wafer diameters from 8 inches to 12 inches. Such a move would greatly reduce the cost of making chips.

Wafer and Dices

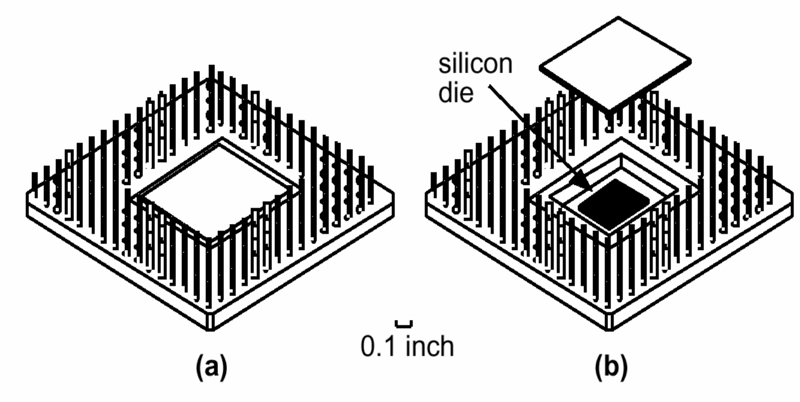


Source: http://cmosedu.com/cmos1/doslasi/doslasi.pdf

Sawing a Wafer into Chips

Figure 9-33
After testing and sawing, the individual chips are picked up by a robotic arm and placed in the package for die bonding. (Photograph courtesy of Intel Corp.)

IC and Die



Chip Packing

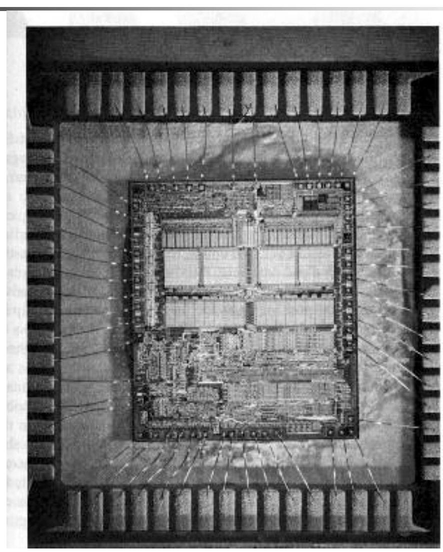
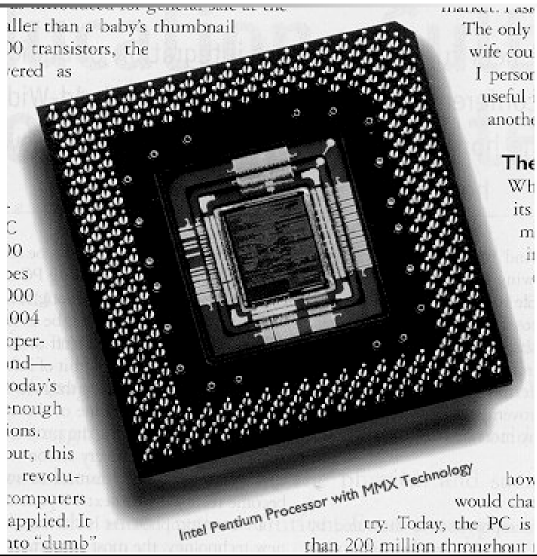
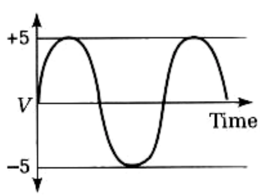


Figure 9-34
Attachment of leads from the Al pads on the periphery of the chip to posts on the package. (Photograph courtesy of Motorola, Inc.)

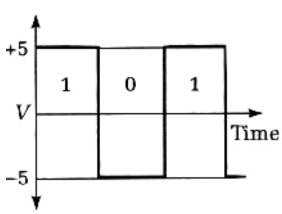
Pentium-MMX with PGA Packaging



Analog v.s. Digital



analog waveform



digital waveform

- Analog system:
  - Inputs and outputs are represented by ***continuous*** values
  - More close to real-world signals
    - Often used as interface circuits
- Digital system:
  - Inputs and outputs are represented by ***discrete*** values
  - Easier to handle and design
    - More tolerable to signal degradation and noise
  - ***Binary digital systems*** form the basis of all hardware design today

Outline

- Introductions
- $\boxed{\text{Number Base Conversions}}$
- Binary Arithmetic
- Binary Codes
- Binary Elements

Binary Numbers

- Decimal numbers → 10 symbols (0, 1, 2, … , 9)

  A10 = $a_{n-1}a_{n-2}\ldots a_1a_0 \,.\, a_{-1}a_{-2}\ldots a_{-m}$

  $$= \sum_{i=-m}^{n-1} a_i * 10^i \qquad 0 \le a_i \le 9$$

  Ex: $(7392)_{10}$ = ***7*** x10^3 + ***3*** x10^2 + ***9*** x10^1 + ***2*** x10^0

- Binary numbers → 2 symbols (0, 1)

  $$A_2 = \sum_{i=-m}^{n-1} a_i * 2^i \qquad a_i = 0,1$$

  Ex: $(11010.11)_2$ = ***1*** x2^4 + ***1*** x2^3 + ***0*** x2^2 + ***1*** x2^1 + ***0*** x2^0 + ***1*** x2^{-1} + ***1*** x2^{-2}

  = $(26.75)_{10}$

Powers of Two

n	2^n	n	2^n	n	2^n
0	1	8	256	16	65536
1	2	9	512	17	131072
2	4	10	1024	18	262144
3	8	11	2048	19	524288
4	16	12	4096	20	1048576
5	32	13	8192	21	2097152
6	64	14	16384	22	4194304
7	128	15	32768	23	8388608

Other Number Bases

- In general, a base–r (radix–r) number system :

$$A_r = \sum_{i=-m}^{n-1} a_i * r^i \qquad 0 \le a_i \le r-1$$

  - Coefficients a_n is multiplied by powers of r
  - Coefficients a_n range in value from 0 to r - 1
- Example :
  - $(4021.2)_5 = \boldsymbol{4} \text{x} 5^3 + \boldsymbol{0} \text{x} 5^2 + \boldsymbol{2} \text{x} 5^1 + \boldsymbol{1} \text{x} 5^0 + \boldsymbol{2} \text{x} 5^{-1}$
    $= (511.4)_{10}$
  - $(127.4)_8 = \boldsymbol{1} \text{x} 8^2 + \boldsymbol{2} \text{x} 8^1 + \boldsymbol{7} \text{x} 8^0 + \boldsymbol{4} \text{x} 8^{-1}$
    $= (87.5)_{10}$
  - $(B65F)_{16} = \boldsymbol{11} \text{x} 16^3 + \boldsymbol{6} \text{x} 16^2 + \boldsymbol{5} \text{x} 16^1 + \boldsymbol{15} \text{x} 16^0$
    $= (46687)_{10}$

Number Base Conversions (1/2)

- Convert $(41)_{10}$ to binary

$41/\mathbf{2}$ =	20	remainder = 1
$20/\mathbf{2}$ =	10	= 0
$10/\mathbf{2}$ =	5	= 0
$5/\mathbf{2}$ =	2	= 1
$2/\mathbf{2}$ =	1	= 0
$1/\mathbf{2}$ =	0	= 1

101001 = answer

$\therefore (41)_{10} = (101001)_2$

- Convert $(0.6875)_{10}$ to binary

$0.6875 \times \mathbf{2} = \boxed{1.}\ 3750$
$0.3750 \times \mathbf{2} = \boxed{0.}\ 7500$
$0.7500 \times \mathbf{2} = \boxed{1.}\ 5000$
$0.5000 \times \mathbf{2} = \boxed{1.}\ 0000$

1011 = answer

$\therefore (0.6875)_{10} = (0.1011)_2$

Number Base Conversions (2/2)

- Convert $(153)_{10}$ to octal

$153/\mathbf{8}$ =	19	remainder = 1
$19/\mathbf{8}$ =	2	= 3
$2/\mathbf{8}$ =	0	= 2

231 = answer

$\therefore (153)_{10} = (231)_8$

- Convert $(0.513)_{10}$ to octal

$0.513 \times \mathbf{8} = \boxed{4.}\ 104$
$0.104 \times \mathbf{8} = \boxed{0.}\ 832$
$0.832 \times \mathbf{8} = \boxed{6.}\ 656$
$0.656 \times \mathbf{8} = \boxed{5.}\ 248$
$0.248 \times \mathbf{8} = \boxed{1.}\ 984$
$0.984 \times \mathbf{8} = \boxed{7.}\ 872$
……

406517 = answer

$\therefore (0.513)_{10} = (0.406517\ldots)_8$

Fast Conversions

- The conversion from and to ***binary***, ***octal***, and ***hexadecimal*** is important but much easier
- To octal : 3 digits per group

  $(10\ 110\ 001\ 101\ 011\ .\ 111\ 100\ 000\ 110)_2 = (26153.7406)_8$

  2 6 1 5 3 7 4 0 6
- To hexadecimal : 4 digits per group

  $(10\ 1100\ 0110\ 1011\ .\ 1111\ 0000\ 0110)_2 = (2C6B.F06)_{16}$

  2 C 6 B F 0 6
- To binary : inverse process

  $(673.124)_8 = (110\ 111\ 011\ .\ 001\ 010\ 100)_2$

  6 7 3 1 2 4

  $(306.D)_{16} = (0011\ 0000\ 0110\ .\ 1101)_2$

  3 0 6 D

Octal & Hexadecimal Numbers

Base 10	Base2	Base 8	Base16	Base10	Base2	Base 8	Base 16
00	0000	00	0	08	1000	10	8
01	0001	01	1	09	1001	11	9
02	0010	02	2	10	1010	12	A
03	0011	03	3	11	1011	13	B
04	0100	04	4	12	1100	14	C
05	0101	05	5	13	1101	15	D
06	0110	06	6	14	1110	16	E
07	0111	07	7	15	1111	17	F

Outline

- Introductions
- Number Base Conversions
- Binary Arithmetic
- Binary Codes
- Binary Elements

Binary Addition

- Binary Addition

  0+0=0 sum of 0 with a carry of 0
  0+1=1 sum of 1 with a carry of 0
  1+0=1 sum of 1 with a carry of 0
  1+1=10 sum of 0 with a carry of 1

$$\begin{array}{r} 101101\ (45) \\ +\ 100111\ (39) \\ \hline 1010100\ (84) \end{array}$$

- Binary Addition with carry

  1+0+0=01 sum of 1 with a carry of 0
  1+0+1=10 sum of 0 with a carry of 1
  1+1+0=10 sum of 0 with a carry of 1
  1+1+1=11 sum of 1 with a carry of 1
  ↑
  carry bit

Binary Subtraction

- Binary Subtraction

  0 – 0 = 0
  1 – 1 = 0
  1 – 0 = 1
  10–1 = 1 (borrow 1 from the higher bit)

  EX :

$$
\begin{array}{rr}
 & 1\ 0\ 1\ 1\ 0\ 1\ (45) \\
- & 1\ 0\ 0\ 1\ 1\ 1\ (39) \\
\hline
 & 0\ 0\ 0\ 1\ 1\ 0\ (6)
\end{array}
$$

Binary Multiplication

- Binary Multiplication

  0 x 0 = 0
  0 x 1 = 0
  1 x 0 = 0
  1 x 1 = 1

  EX :

$$
\begin{array}{rr}
 & 1\ 0\ 1\ (5) \\
\times & 1\ 0\ 1\ (5) \\
\hline
 & 1\ 0\ 1 \\
 & 0\ 0\ 0 \\
 & 1\ 0\ 1 \\
\hline
 & 1\ 1\ 0\ 0\ 1\ (25)
\end{array}
$$

Binary Division

- Binary Division

  EX : 110 ÷ 10 = 11

$$
\begin{array}{r}
11 \\
10\overline{)\,110} \\
\underline{10} \\
10 \\
\underline{10} \\
00
\end{array}
$$

Complements

- Unsigned binary arithmetic is quite similar to decimal operations
- But how about the negative numbers ???
  - Use the ***complements*** of positive numbers
- Complements can simplify the subtraction operation and logical manipulation
- Two types of complements :
  - Diminished radix complement
  - Radix complement

Diminished Radix Complement

- Given a number N in base r having n digits, the (r-1)'s complement of N is defined as $(r^n-1)-N$
  - The 9's complement of $(546700)_{10}$ is 999999 – 546700 = 453299
  - The 9's complement of $(012398)_{10}$ is 999999 – 012398 = 987601
  - The 1's complement of $(1011000)_2$ is 0100111
  - The 1's complement of $(0101101)_2$ is 1010010

bit inverse only !!

Radix Complement

- The r's complement of an n-digit number N in base r is defined as r^n-N, for $N \neq 0$ and 0 for N=0
- Equal to its (r-1)'s complement added by 1
  - The 10's complement of 012398 is 987602
  - The 10's complement of 246700 is 753300
  - The 2's complement of 1101100 is 0010100
  - The 2's complement of 0110111 is 1001001

bit inverse and
added with 1

Subtraction with Complements

- The subtraction of two n-digit unsigned numbers M – N in base r can be done as follows：

  (1)：Add the minuend, M, to the ***r's complement*** of the subtrahend, N.

  $M+(r^n – N)=M – N + r^n$

  (2)：If $M \geq N$, the sum will produce an end carry, r^n, which can be ***discarded***; what is left is the result M – N

  (3)：if M < N, the sum does not produce an end carry and is equal to $r^n – (N – M)$, which is the r's complement of (N – M). To obtain the answer in a familiar form , take the r's complement of the sum and placed a ***negative sign*** in front

Examples of Subtraction

Given two binary numbers X = 1010100 and Y = 1000011

- X – Y (use 2's complement)

  2's complement of Y

  = 0111100 + 1

  = 0111101

  $\Rightarrow$ X – Y = X + Y'

  $$\begin{array}{r} 1010100 \\ +0111101 \\ \hline 1\,0010001 \end{array}$$

  carry is discarded

  $\therefore$ X – Y = 0010001

- Y – X (use 2's complement)

  2's complement of X

  = 0101011 + 1

  = 0101100

  $\Rightarrow$ Y – X = Y + X'

  $$\begin{array}{r} 1000011 \\ +0101100 \\ \hline 1101111 \end{array}$$

  no end carry generated

  $\therefore$ Y – X = 1101111

Signed Binary Numbers

- Signed 2's complement
⇒ use 2's comp.
- Signed 1's complement
⇒ use 1's comp.
- Signed magnitude
⇒ first digit: sign
others: magnitude

Decimal	Signed 2's complement	Signed 1's complement	Signed magnitude
+7	0111	0111	0111
+6	0110	0110	0110
+5	0101	0101	0101
+4	0100	0100	0100
+3	0011	0011	0011
+2	0010	0010	0010
+1	0001	0001	0001
+0	0000	0000	0000
- 0	—	1111	1000
-1	1111	1110	1001
-2	1110	1101	1010
-3	1101	1100	1011
-4	1100	1011	1100
-5	1011	1010	1101
-6	1010	1001	1110
-7	1001	1000	1111
-8	1000	—	—

Arithmetic Addition/Subtraction

- The negative numbers are in ***2's complement*** form
- If the sum is negative, it's also in 2's complement form

$$\begin{array}{rr} +6 & 00000110 \\ +13 & 00001101 \\ \hline +19 & 00010011 \end{array} \qquad \begin{array}{rr} -6 & 11111010 \\ +13 & 00001101 \\ \hline +7 & 00000111 \end{array}$$

$$\begin{array}{rr} +6 & 00000110 \\ -13 & 11110011 \\ \hline -7 & 11111001 \end{array} \qquad \begin{array}{rr} -6 & 11111010 \\ -13 & 11110011 \\ \hline -19 & 11101101 \end{array}$$

- For subtraction, take 2's complement of the subtrahend

$$(\pm A) - (+B) = (\pm A) + (-B)$$
$$(\pm A) - (-B) = (\pm A) + (+B)$$

Outline

- Introductions
- Number Base Conversions
- Binary Arithmetic
- Binary Codes
- Binary Elements

BCD Code

- BCD = Binary Coded Decimal
- ***4 bits*** for one digit
  Ex: $(185)_{10}$
  $= (0001\ 1000\ 0101)_{BCD}$
  $= (10111001)_2$
- ***1010 to 1111*** are not used and have no meaning in BCD
- Can perform arithmetic operations directly with decimal numbers in digital systems

Decimal symbol	BCD Digit
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

BCD Addition

- When the binary sum is equal to or less than 1001 (without a carry), the corresponding BCD digit is correct
- Otherwise, the addition of ***6 = (0110)$_2$*** can convert it to the correct digit and also produce the required carry

4	0100
+ 5	0101
9	1001

4	0100
+ 8	1000
12	1100
	+ 0110
	10010

8	1000
+ 9	1001
17	10001
	+ 0110
	10111

$9 \rightarrow (0\ 1001)_{BCD}$

$10 \rightarrow (1\ 0000)_{BCD}$

$10+6 = 16 = (10000)_2$

$\therefore$ The addition of 6 can convert to correct digit

Other Decimal Codes

weight

DECIMAL DIGIT	BCD (8421)	2421	EXCESS-3	84-2-1
0	0000	0000	0011	0000
1	0001	0001	0100	0111
2	0010	0010	0101	0110
3	0011	0011	0110	0101
4	0100	0100	0111	0100
5	0101	1011	1000	1011
6	0110	1100	1001	1010
7	0111	1101	1010	1001
8	1000	1110	1011	1000
9	1001	1111	1100	1111
Unused bit combinations	1010	0101	0000	0001
	1011	0110	0001	0010
	1100	0111	0010	0011
	1101	1000	1101	1100
	1110	1001	1110	1101
	1111	1010	1111	1110

Gray Code

Only change one bit when going from one number to the next !!

Gray code	Decimal equivalent	Gray code	Decimal equivalent
0000	0	1100	8
0001	1	1101	9
0011	2	1111	10
0010	3	1110	11
0110	4	1010	12
0111	5	1011	13
0101	6	1001	14
0100	7	1000	15

ASCII Character Code

American Standard Code for Information Interchange (ASCII)

	$b_7 b_6 b_5$							
$b_4 b_3 b_2 b_1$	**000**	**001**	**010**	**011**	**100**	**101**	**110**	**111**
0000	NUL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	∧	n	~
1111	SI	US	/	?	O	—	o	DEL

Error-Detecting Code

- To detect error in data communication, an extra ***parity*** bit is often added
- Just count the total number of “1” can decide the status of parity bit
  - Even parity = 1 → the number of “1” is even (include parity)
  - Odd parity = 1 → the number of “1” is odd (include parity)
- If the received number of “1” does not match the parity bit, an error occurs

	with even parity	with odd parity
ASCII A = 1000001	0 1000001	1 1000001
ASCII T = 1010100	1 1010100	0 1010100

parity bit

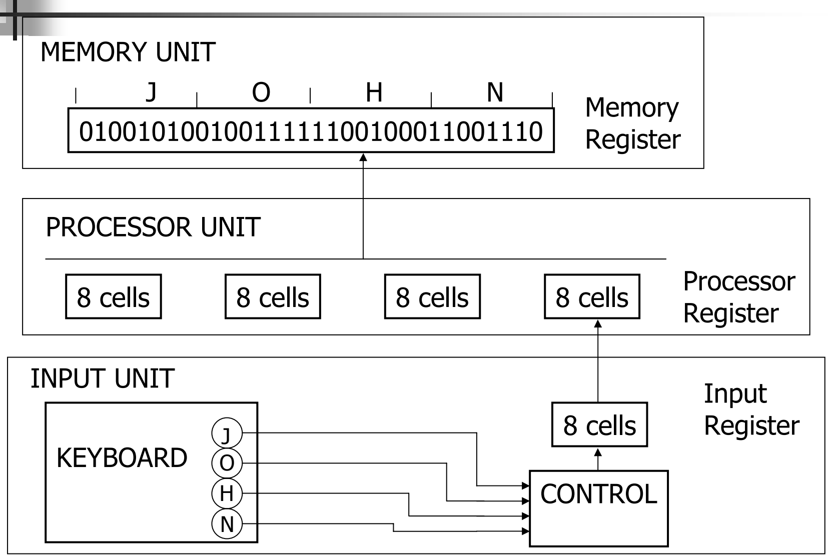
Outline

- Introductions
- Number Base Conversions
- Binary Arithmetic
- Binary Codes
- Binary Elements

Binary Elements

- Binary cell
  - Possess two stable states and store one bit of information
- Registers
  - A group of binary cells
  - A register with n cells can store any discrete quantity of n-bit information
- Binary logic
  - Define the operations with variables that take two discrete values
  - The operations are implemented by ***logic gates***
- Register transfers (to where) and data operations (do what) form the digital systems

Register Transfer of Information

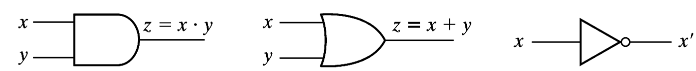


Logical Operations

- Truth tables of logical operations:

AND			OR			NOT	
X	Y	X • Y	X	Y	X + Y	X	X'
0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0
1	0	0	1	0	1		
1	1	1	1	1	1		

- Logic gates: electronic circuits that perform the corresponding logical operations

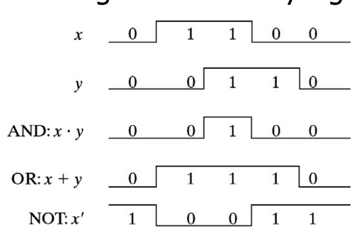


(a) Two-input AND gate (b) Two-input OR gate (c) NOT gate or inverter

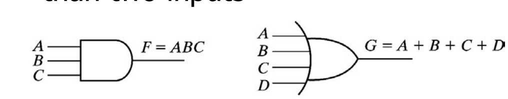
Fig. 1-4 Symbols for digital logic circuits

Properties of Logic Gates

- Logic gates handle binary signals and also generate binary signals



- AND and OR gates may have more than two inputs



(a) Three-input AND gate (b) Four-input OR gate

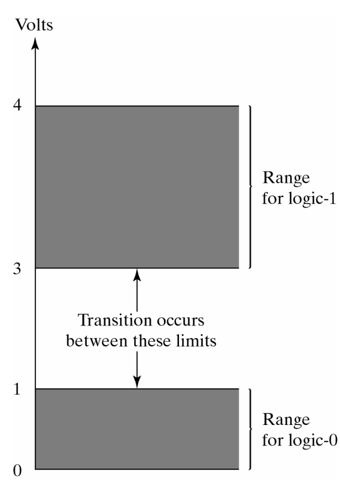


Fig. 1-3 Example of binary signals