CSEE 3827: Fundamentals of Computer Systems

Lecture 1: Introduction

1/20/10

Agenda

- Administrative details
- Course introduction
- Information representation and definitions

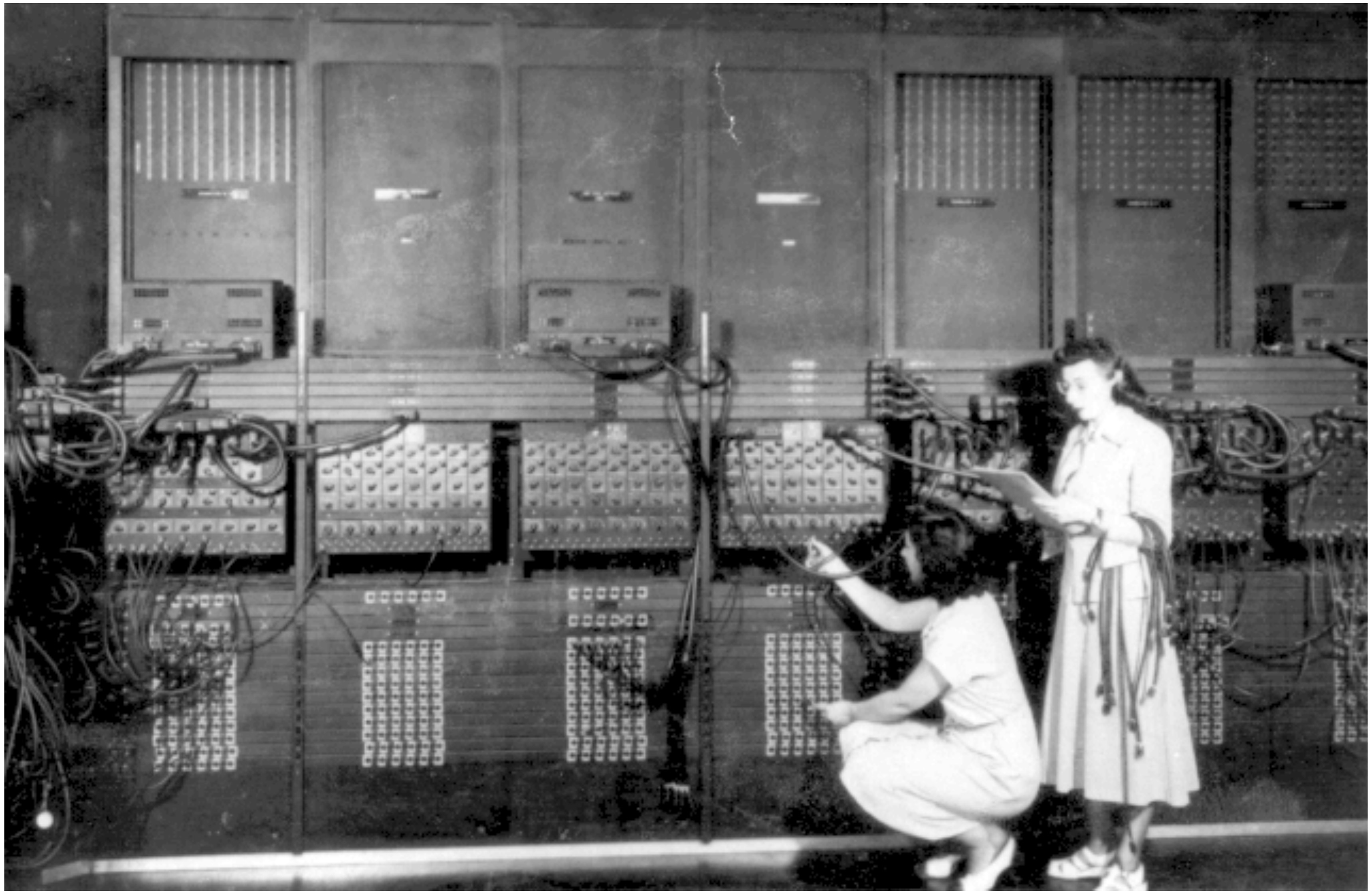
Course website

http://www.cs.columbia.edu/~martha/courses/3827/sp10/

Agenda

- Administrative details
- Course introduction
- Information representation and definitions

What does this ...



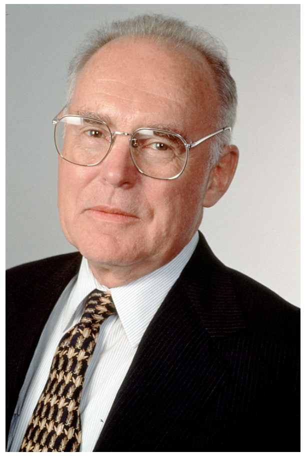
[Source: http://ftp.arl.army.mil/~mike/comphist]

... have in common with this?

growth in performance = growth in raw resources + system design innovation

ENIAC (1946)	Intel Larrabee (projected 2010)	
5,000 operations per second	2,000,000,000,000 operations per second	400,000,000x faster
8.5’ x 3’ x 80’ (2040 ft^3)	49.5 mm^2	1,167,000,000x smaller
$500,000	~$300	1666x cheaper

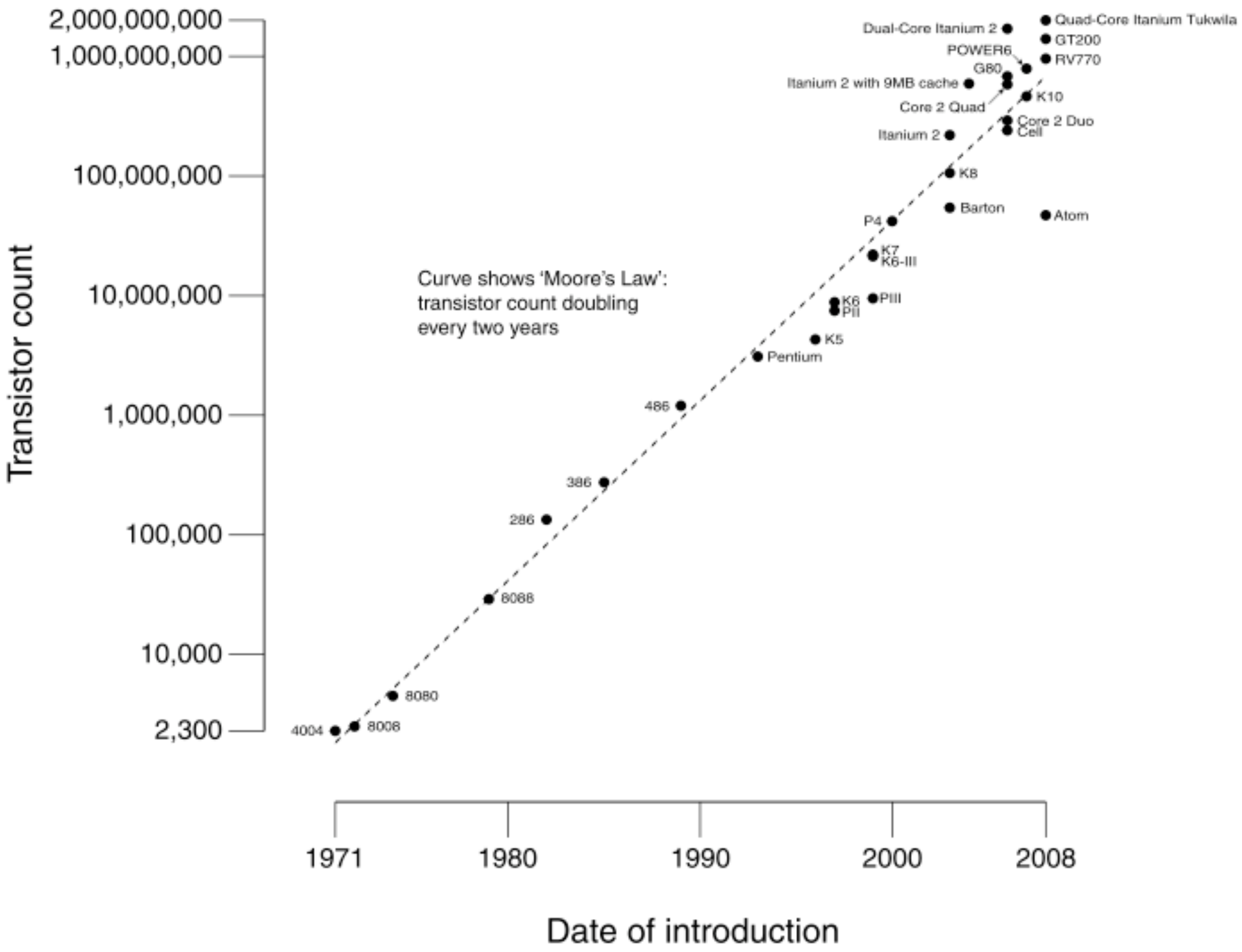
growth in performance = growth in raw resources + system design innovation



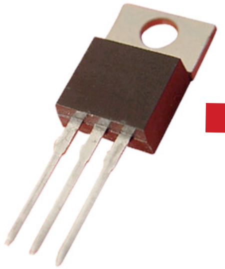
Gordon Moore
co-founder of Intel

Moore’s Law:
Density of transistors
doubles every two years

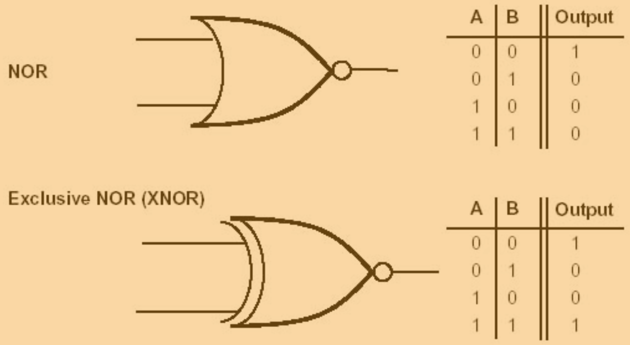
CPU Transistor Counts 1971-2008 & Moore’s Law



growth in performance = growth in raw resources + system design innovation



transistors



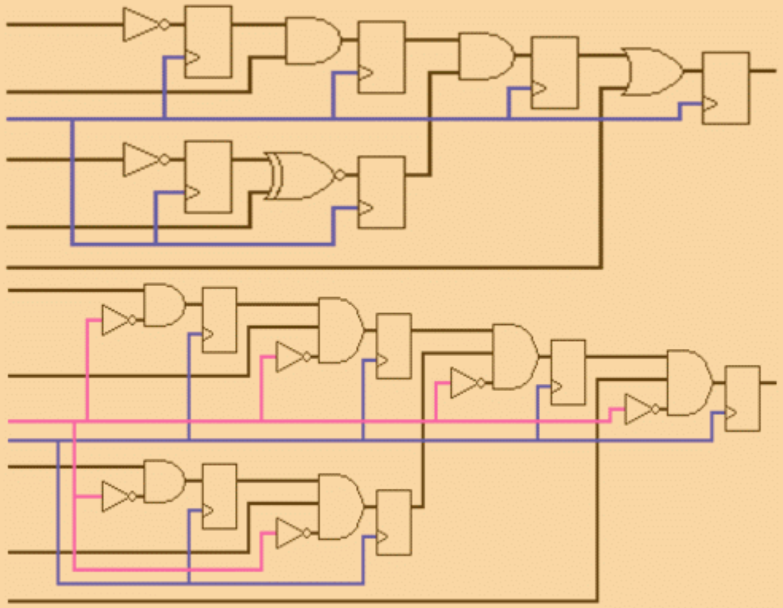
NOR

A	B	Output
0	0	1
0	1	0
1	0	0
1	1	0

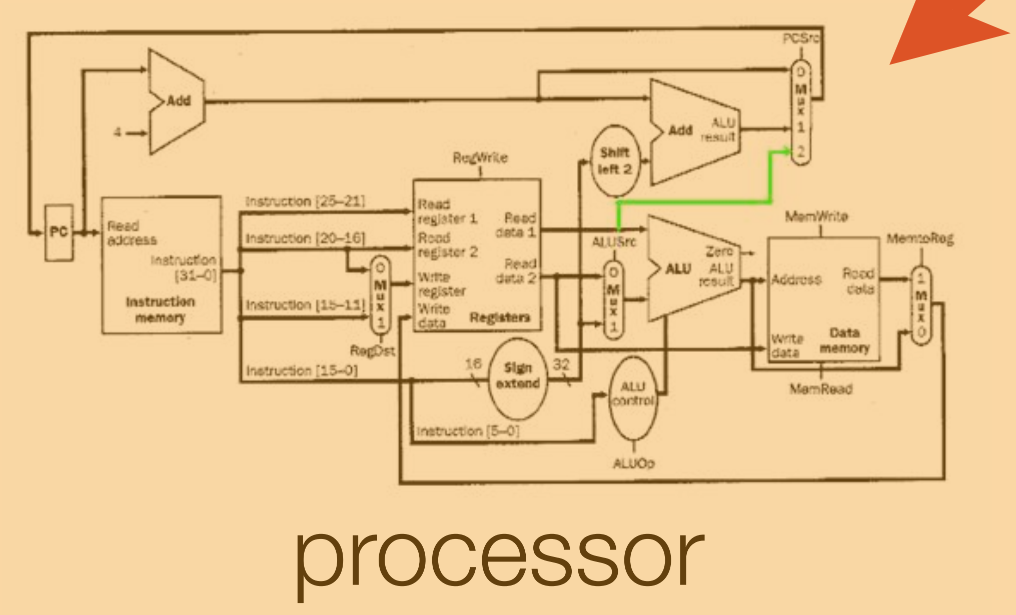
Exclusive NOR (XNOR)

A	B	Output
0	0	1
0	1	0
1	0	0
1	1	1

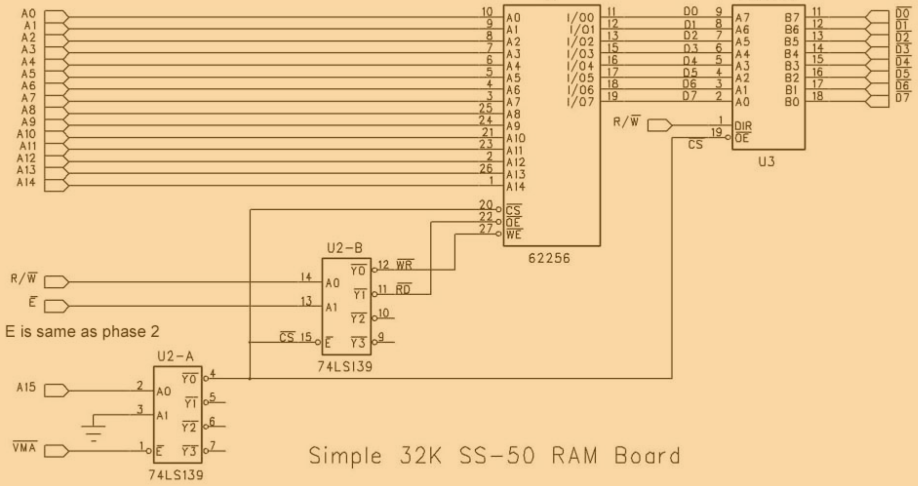
logic gates



logic circuits



processor



memory

Agenda

- Administrative details
- Course introduction
- Information representation and definitions

Number systems: Base 10 (Decimal)

- 10 digits = {0,1,2,3,4,5,6,7,8,9}
- example: 4537.8 = $(4537.8)_{10}$

$$\begin{array}{cccccccc} 4 & & 5 & & 3 & & 7 & . & 8 & & \\ \times 10^{3} & & \times 10^{2} & & \times 10^{1} & & \times 10^{0} & & \times 10^{-1} & & \\ \hline 4000 & + & 500 & + & 40 & + & 7 & + & .8 & = & 4537.8 \end{array}$$

Number systems: Base 2 (Binary)

- 2 digits = {0,1}
- example: 1011.1 = $(1011.1)_2$

$$\begin{array}{ccccccccc} 1 & & 0 & & 1 & & 1 & . & 1 \\ \times 2^3 & & \times 2^2 & & \times 2^1 & & \times 2^0 & & \times 2^{-1} \\ \hline 8 & + & 0 & + & 2 & + & 1 & + & .5 \end{array} = (11.5)_{10}$$

Number systems: Base 8 (Octal)

- 8 digits = {0,1,2,3,4,5,6,7}
- example: $(2365.2)_8$

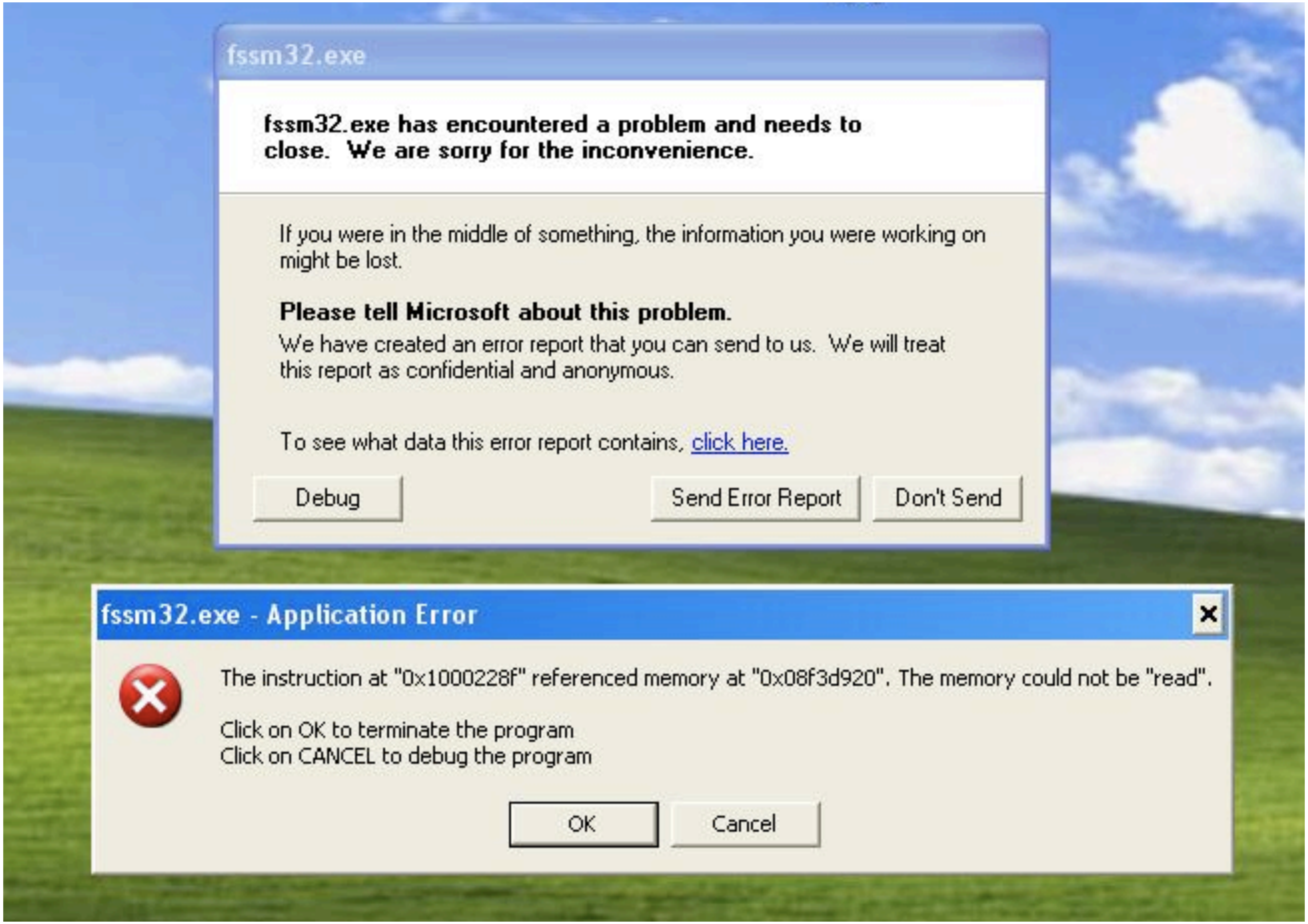
$$\begin{array}{cccccccccc}
2 & & 3 & & 6 & & 5 & . & 2 & \\
\times\ 8^{3} & & \times\ 8^{2} & & \times\ 8^{1} & & \times\ 8^{0} & & \times\ 8^{-1} & \\
\hline
1024 & + & 192 & + & 48 & + & 5 & + & .25 & = (1269.25)_{10}
\end{array}$$

Number systems: Base 16 (Hexadecimal)

- 16 digits = {0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F}
- example: $(26BA)_{16}$ [alternate notation for hex: 0x26BA]

$$\begin{array}{ccccccc} 2 & & 6 & & B & & A \\ \times\ 16^{3} & & \times\ 16^{2} & & \times\ 16^{1} & & \times\ 16^{0} \\ \hline 8192 & + & 1536 & + & 176 & + & 10 \end{array} = (9914)_{10}$$

Hexadecimal (or hex) is often used for addressing



fssm32.exe

fssm32.exe has encountered a problem and needs to close. We are sorry for the inconvenience.

If you were in the middle of something, the information you were working on might be lost.

Please tell Microsoft about this problem.

We have created an error report that you can send to us. We will treat this report as confidential and anonymous.

To see what data this error report contains, click here.

fssm32.exe - Application Error

The instruction at "0x1000228f" referenced memory at "0x08f3d920". The memory could not be "read".

Click on OK to terminate the program
Click on CANCEL to debug the program

Number ranges

- Map infinite numbers onto finite representation for a computer

- How many numbers can I represent with ...

  ... 5 digits in decimal? 10^5 possible values

  ... 8 binary digits? 2^8 possible values

  ... 4 hexadecimal digits? 16^4 possible values

Need a bigger range?

- Change the encoding.
- Floating point (used to represent very large numbers in a compact way)
  - A lot like scientific notation: 5.4×10^{5} exponent

    mantissa
  - Except that it is binary: $\underline{1001} \times 2^{\underline{1011}}$

What about negative numbers?

- Change the encoding.
  - Sign and magnitude
  - Ones compliment
  - Twos compliment

Sign and magnitude

- Most significant bit is sign
- Rest of bits are magnitude

$$0110 = (6)_{10} \qquad 1110 = (-6)_{10}$$

- Two representations of zero

$$0000 = (0)_{10} \qquad 1000 = (-0)_{10}$$

Ones compliment

- Compliment bits in positive value to create negative value
- Most significant bit still a sign bit

$0110 = (6)_{10}$ $1001 = (-6)_{10}$

- Two representations of zero

$0000 = (0)_{10}$ $1111 = (-0)_{10}$

Twos compliment

- Compliment bits in positive value and add 1 to create negative value
- Most significant bit still a sign bit

$0110 = (6)_{10}$ $1001 + 1 = 1010 = (-6)_{10}$

- One representation of zero

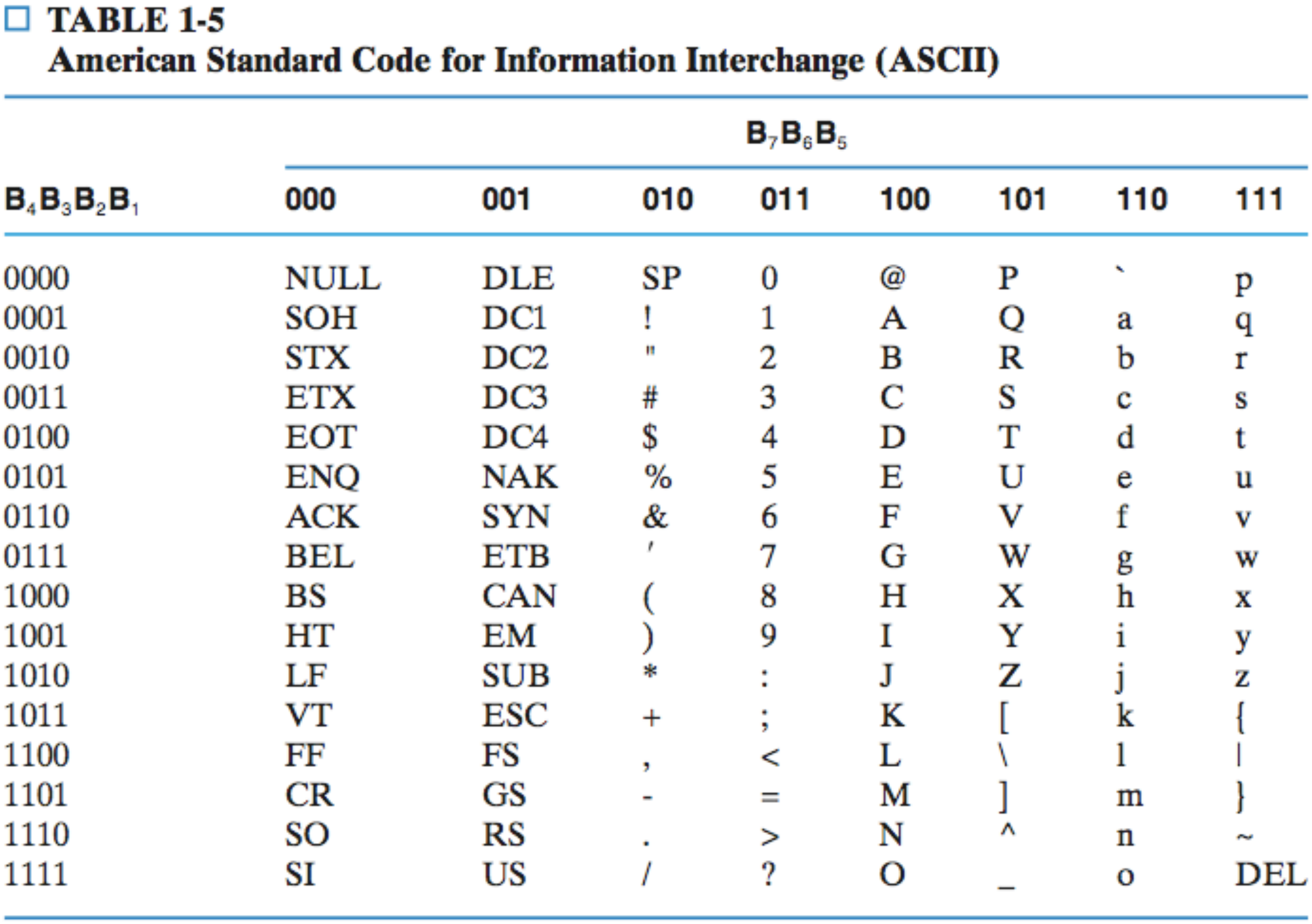
$0000 = (0)_{10}$ $1000 = (-8)_{10}$ $1111 = (-1)_{10}$

- One more negative number than positive

MIN: $1000 = (-8)_{10}$ MAX: $0111 = (7)_{10}$

How about letters?

- Change the encoding.



☐ **TABLE 1-5**
American Standard Code for Information Interchange (ASCII)

	$B_7B_6B_5$							
$B_4B_3B_2B_1$	000	001	010	011	100	101	110	111
0000	NULL	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

Gray code

Binary numeric encoding where successive numbers differ by only 1 bit

value	BCD	*# bit flips*	Gray	*# bit flips*
0	0 0 0	*3*	0 0 0	*1*
1	0 0 1	*1*	0 0 1	*1*
2	0 1 0	*2*	0 1 1	*1*
3	0 1 1	*1*	0 1 0	*1*
4	1 0 0	*3*	1 1 0	*1*
5	1 0 1	*1*	1 1 1	*1*
6	1 1 0	*2*	1 0 1	*1*
7	1 1 1	*1*	1 0 0	*1*

Some definitions

- bit = a binary digit e.g., 1 or 0
- byte = 8 bits e.g., 01100100
- word = a group of bytes

  a 16-bit word = 2 bytes e.g., 1001110111000101

  a 32-bit word = 4 bytes e.g., 100111011100010101110111000101

Next class: binary logic, logic gates