Lecture 4: Continued: Bits, bytes, binary numbers, and the representation of information

- **computers represent, process, store, copy, and transmit everything as numbers**
  - hence "digital computer"

- **the numbers can represent anything**
  - not just numbers that you might do arithmetic on

- **the meaning depends on context**
  - as well as what the numbers ultimately represent
  - e.g., numbers coming to your computer or phone from your wi-fi connection could be email, movies, music, documents, apps, Zoom meeting, ...

Some things are intrinsically discrete / digital

- another kind of conversion
  - letters are converted into numbers when you type on a keyboard
  - the letters are stored (a Word document), retrieved (File/Open...), processed (paper is revised), transmitted (submitted by email), printed on paper
- letters and other symbols are inherently discrete
- encoding them as numbers is just assigning a numeric value to each one, without any intrinsic meaning
- what letters and other symbols are included?
- how many digits/letter?
  - determined by how many symbols there are
  - how do we disambiguate if symbols have different lengths?
- how do we decide whose encoding to use?
- the representation is arbitrary
- but everyone has to agree on it
  - if they want to work together

ASCII: American Standard Code for Information Interchange

- an arbitrary but agreed-upon representation for USA
- widely used everywhere

<table>
<tr><td>32</td><td>space</td><td>33</td><td>!</td><td>34</td><td>"</td><td>35</td><td>#</td><td>36</td><td>$</td><td>37</td><td>%</td><td>38</td><td>&</td><td>39</td><td>'</td></tr>
<tr><td>40</td><td>(</td><td>41</td><td>)</td><td>42</td><td>*</td><td>43</td><td>+</td><td>44</td><td>,</td><td>45</td><td>-</td><td>46</td><td>.</td><td>47</td><td>/</td></tr>
<tr><td>48</td><td>0</td><td>49</td><td>1</td><td>50</td><td>2</td><td>51</td><td>3</td><td>52</td><td>4</td><td>53</td><td>5</td><td>54</td><td>6</td><td>55</td><td>7</td></tr>
<tr><td>56</td><td>8</td><td>57</td><td>9</td><td>58</td><td>:</td><td>59</td><td>;</td><td>60</td><td><</td><td>61</td><td>=</td><td>62</td><td>></td><td>63</td><td>?</td></tr>
<tr><td>64</td><td>@</td><td>65</td><td>A</td><td>66</td><td>B</td><td>67</td><td>C</td><td>68</td><td>D</td><td>69</td><td>E</td><td>70</td><td>F</td><td>71</td><td>G</td></tr>
<tr><td>72</td><td>H</td><td>73</td><td>I</td><td>74</td><td>J</td><td>75</td><td>K</td><td>76</td><td>L</td><td>77</td><td>M</td><td>78</td><td>N</td><td>79</td><td>O</td></tr>
<tr><td>80</td><td>P</td><td>81</td><td>Q</td><td>82</td><td>R</td><td>83</td><td>S</td><td>84</td><td>T</td><td>85</td><td>U</td><td>86</td><td>V</td><td>87</td><td>W</td></tr>
<tr><td>88</td><td>X</td><td>89</td><td>Y</td><td>90</td><td>Z</td><td>91</td><td>[</td><td>92</td><td>\</td><td>93</td><td>]</td><td>94</td><td>^</td><td>95</td><td>_</td></tr>
<tr><td>96</td><td>`</td><td>97</td><td>a</td><td>98</td><td>b</td><td>99</td><td>c</td><td>100</td><td>d</td><td>101</td><td>e</td><td>102</td><td>f</td><td>103</td><td>g</td></tr>
<tr><td>104</td><td>h</td><td>105</td><td>i</td><td>106</td><td>j</td><td>107</td><td>k</td><td>108</td><td>l</td><td>109</td><td>m</td><td>110</td><td>n</td><td>111</td><td>o</td></tr>
<tr><td>112</td><td>p</td><td>113</td><td>q</td><td>114</td><td>r</td><td>115</td><td>s</td><td>116</td><td>t</td><td>117</td><td>u</td><td>118</td><td>v</td><td>119</td><td>w</td></tr>
<tr><td>120</td><td>x</td><td>121</td><td>y</td><td>122</td><td>z</td><td>123</td><td>{</td><td>124</td><td>|</td><td>125</td><td>}</td><td>126</td><td>~</td><td>127</td><td>del</td></tr>
</table>

`00010000 space   00010001 !   00010010 "   00010101 #   ...`

Hexadecimal notation

- binary numbers are bulky
- hexadecimal notation is a shorthand
- it combines 4 bits into a single digit, written in base 16
  - a more compact representation of the same information
- hex uses the symbols A B C D E F for the digits 10 .. 15

0 1 2 3 4 5 6 7 8 9 A B C D E F

0	0000	1	0001	2	0010	3	0011
4	0100	5	0101	6	0110	7	0111
8	1000	9	1001	A	1010	B	1011
C	1100	D	1101	E	1110	F	1111

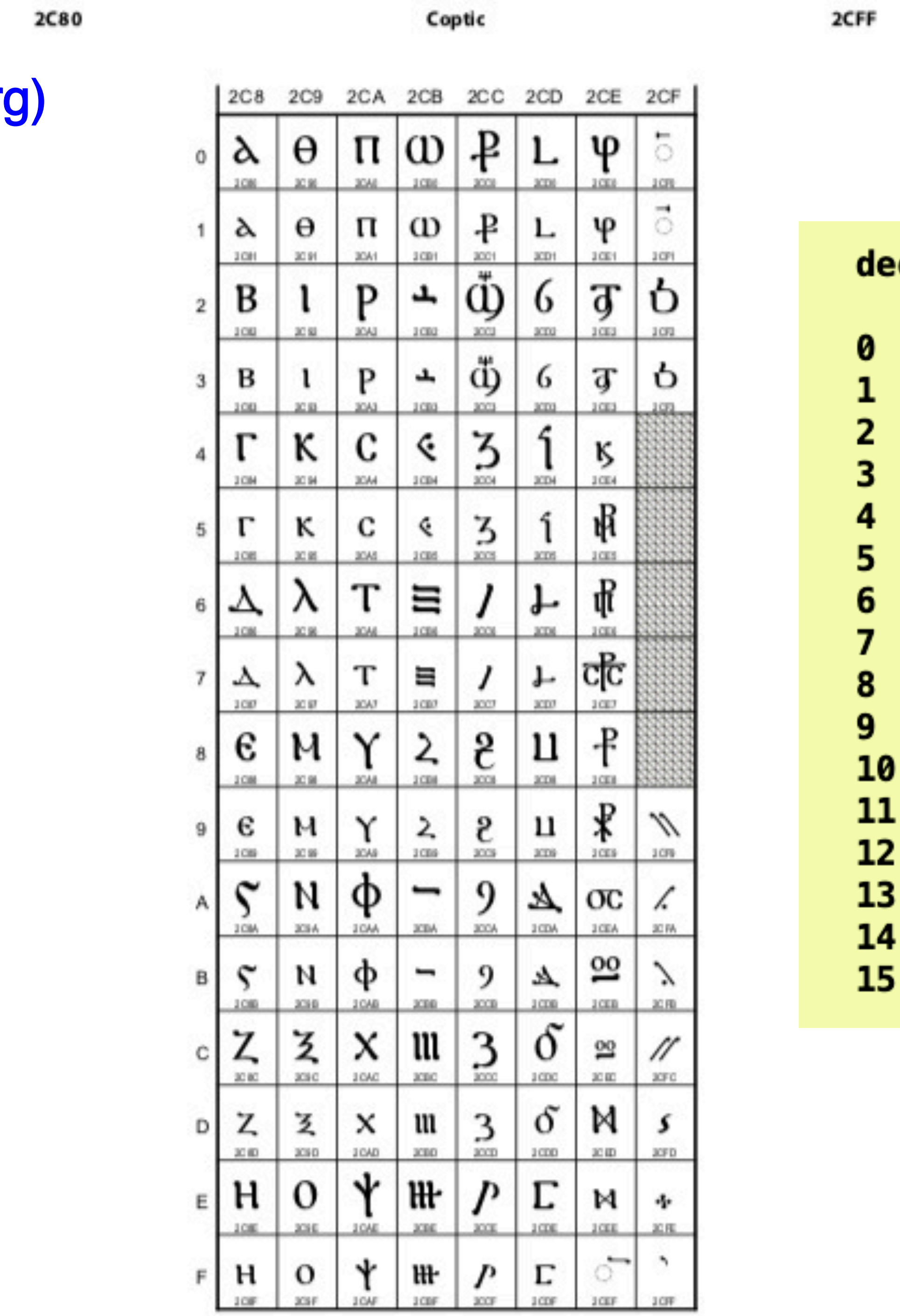
Decimal, binary, hex

dec	bin	hex
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

ASCII, using hexadecimal numbers

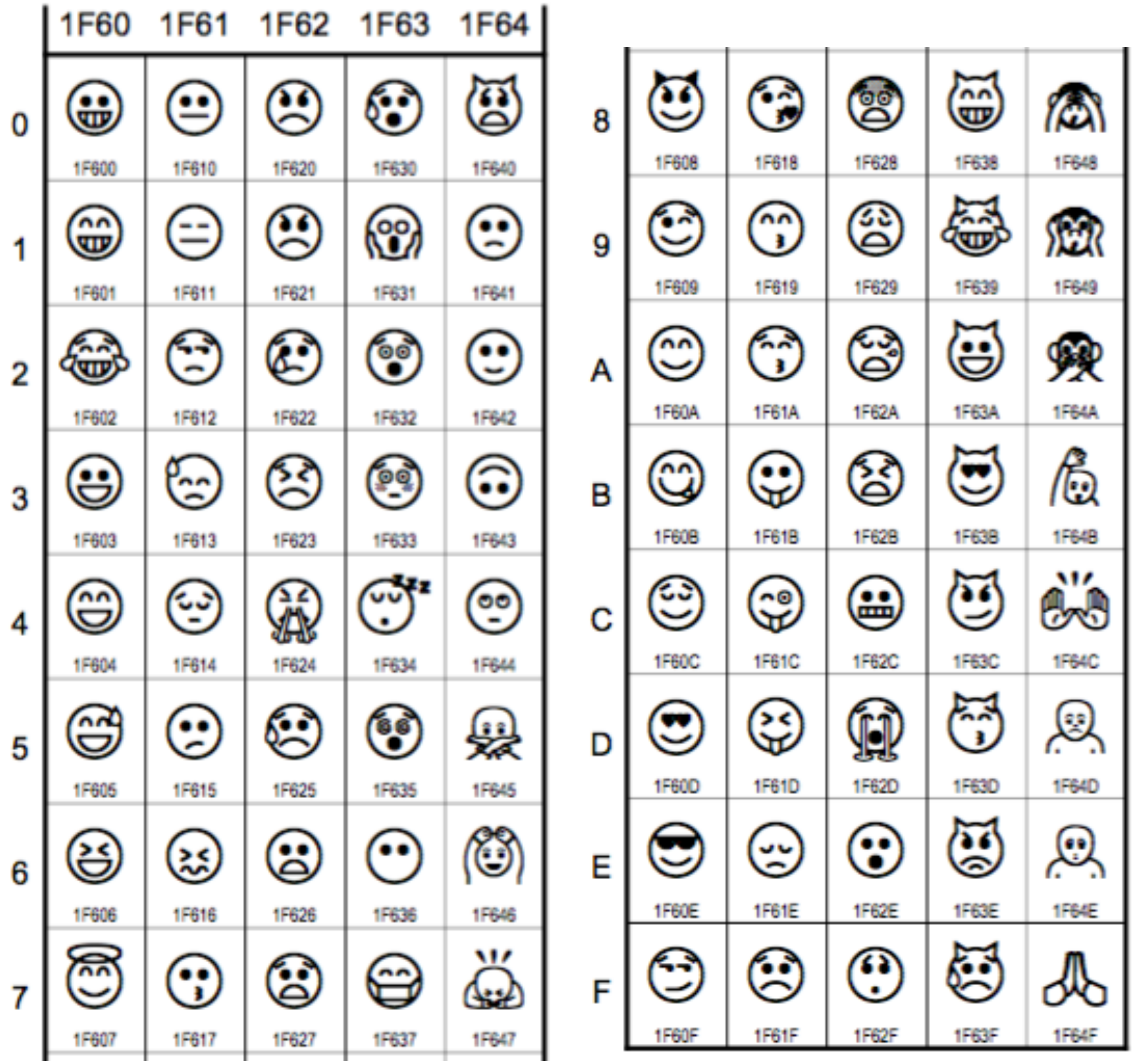
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SPC	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

Coptic (unicode.org)



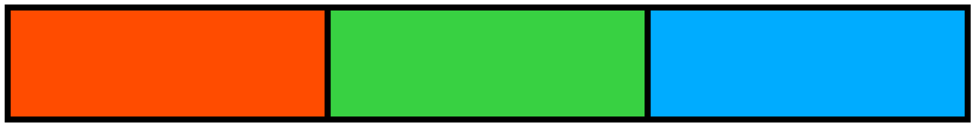
dec	bin	hex
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Emoji


1F60 1F61 1F62 1F63 1F64
0
1F600 1F610 1F620 1F630 1F640
1
1F601 1F611 1F621 1F631 1F641
2
1F602 1F612 1F622 1F632 1F642
3
1F603 1F613 1F623 1F633 1F643
4
1F604 1F614 1F624 1F634 1F644
5
1F605 1F615 1F625 1F635 1F645
6
1F606 1F616 1F626 1F636 1F646
7
1F607 1F617 1F627 1F637 1F647
8
1F608 1F618 1F628 1F638 1F648
9
1F609 1F619 1F629 1F639 1F649
A
1F60A 1F61A 1F62A 1F63A 1F64A
B
1F60B 1F61B 1F62B 1F63B 1F64B
C
1F60C 1F61C 1F62C 1F63C 1F64C
D
1F60D 1F61D 1F62D 1F63D 1F64D
E
1F60E 1F61E 1F62E 1F63E 1F64E
F
1F60F 1F61F 1F62F 1F63F 1F64F

Color

- TV & computer screens use Red-Green-Blue (RGB) model



- each color is a combination of red, green, blue components
  – R+G = yellow, R+B = magenta, B+G = cyan, R+G+B = white
- for computers, color of a pixel is usually specified by three numbers giving amount of each color, on a scale of 0 to 255
- this is often expressed in hexadecimal so the three components can be specified separately (in effect, as bit patterns)
  – 000000 is black, FFFFFF is white
- printers, etc., use cyan-magenta-yellow[-black] (CMY[K])

Approximations using 2^n

2^24 = 2^4 * 2^20

= 16 * 1,000,000 (16,777,216)

2^32 = 2^2 * 2^30

= 4 * 1,000,000,000 (4,294,967,296)

2^64 = 2^4 * 2^60

= 16 * 1,000,000,000,000,000,000

(18,446,744,073,709,551,616)

A very important idea

- **number of items and number of digits are tightly related:**
  - one determines the other
  - maximum number of different items = base $^{\text{number of digits}}$
  - e.g., 9-digit SSN: 10^9 = 1 billion possible numbers

  - e.g., to represent up to 100 “characters”: 2 digits is enough
  - but for 1000 characters, we need 3 digits

  - the same for bits: 9 bits can represent up to 2^9 = 512 items

- **interpretation depends on context**
  - without knowing that, we can only guess what numbers mean

Things to remember

- **digital devices represent everything as numbers**
  - discrete values, not continuous or infinitely precise
- **all modern digital devices use binary numbers (base 2)**
  - instead of decimal (base 10)
- **it's all bits at the bottom**
  - a bit is a "binary digit", that is, a number that is either 0 or 1
  - computers ultimately represent and process everything as bits
- **groups of bits represent larger things**
  - numbers, letters, words, names, pictures, sounds, instructions, ...
  - the interpretation of a group of bits depends on their context
  - the representation is arbitrary; standards (often) define what it is
- **the number of digits used in the representation determines how many different things can be represented**
  - number of values = $\text{base}^{\text{number of digits}}$
  - e.g., 10^2 , 2^{10}