SCS1305 / IS1202
Computer Systems

Data Representation

Dr. Ajantha Atukorale
aja@ucsc.cmb.ac.lk

Data Representation In Computers

Data Representation

- Data Representation → Representing **Numbers**, representing **Characters** and representing **Symbols**.
- Data Representation → Can be **Quantitative** or **Qualitative**

- Quantitative data can be **counted**, **measured**, and expressed using **numbers**.
- Quantitative data can be **Integers** or **Non-Integers** and they can be **signed** or **unsigned** as well.

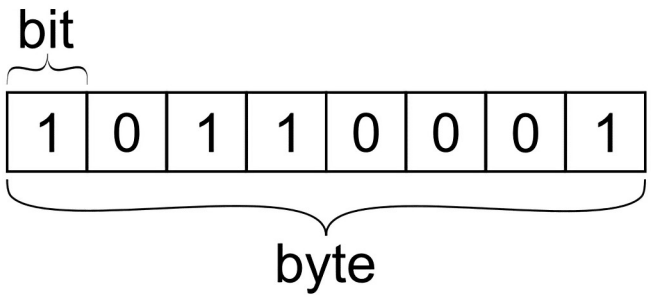
- Qualitative data is **descriptive** and **conceptual**.
- Qualitative data can be **categorized based on** traits (attributes) and characteristics

What is a bit

- A **bit** is a binary digit, the smallest increment of data on a machine. A **bit** can hold only one of two values: **0** or **1**
- Because **bits** are so small, you rarely work with information one **bit** at a time.

What is a Byte

- **Byte** is an abbreviation for "binary term". A single byte is composed of 8 consecutive bits capable of storing a single character



Storage Hierarchy

- 8 Bits = 1 Byte
- 1024 Bytes = 1 Kilobyte (KB)
- 1024 KB = 1 Megabyte (MB)
- 1024 MB = 1 Gigabyte (GB)

Prefix	Symbol(s)	Power of 10	Power of 2
kilo-	k or K **	10^3	2^{10}
mega-	M	10^6	2^{20}
giga-	G	10^9	2^{30}
tera-	T	10^{12}	2^{40}
peta-	P	10^{15}	2^{50}
exa-	E	10^{18} *	2^{60}
zetta-	Z	10^{21} *	2^{70}
yotta-	Y	10^{24} *	2^{80}
* Not generally used to express data speed			
** k = 10^3 and K = 2^{10}			

Word

- A **word** is the natural unit of data used by a particular processor design.
- A word is a **fixed-sized piece of data** handled as a **unit by the instruction set** or the hardware of the processor.
- The **number of bits in a word** (the word size, word width, or word length) is an important characteristic of any specific processor design or computer architecture.

Numbering System

- Decimal System
  - Alphabet = { 0,1,2,3,4,5,6,7,8,9 }
- Octal System
  - Alphabet = { 0,1,2,3,4,5,6,7 }
- Hexadecimal System
  - Alphabet = { 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F }
- Binary System
  - Alphabet = { 0,1 }

Leading Zeros

- A leading zero is any 0 digit that comes **before** the **first nonzero digit** in a number string in positional notation.
- Leading Zeros are **NOT significant**.
- They are nothing more than **place holders**.

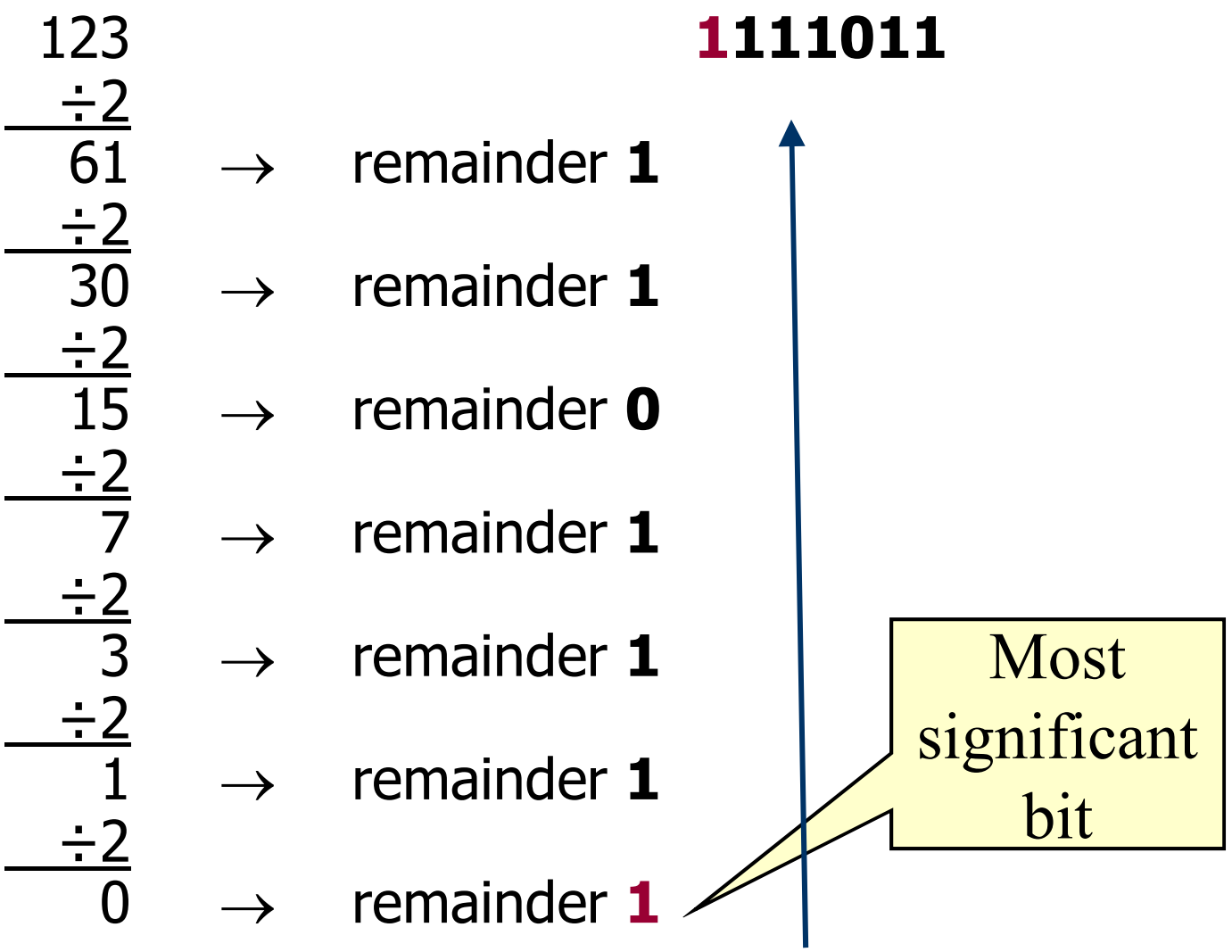
- Q: What is the meaning of **007** in James Bond series?

Converting decimal to binary

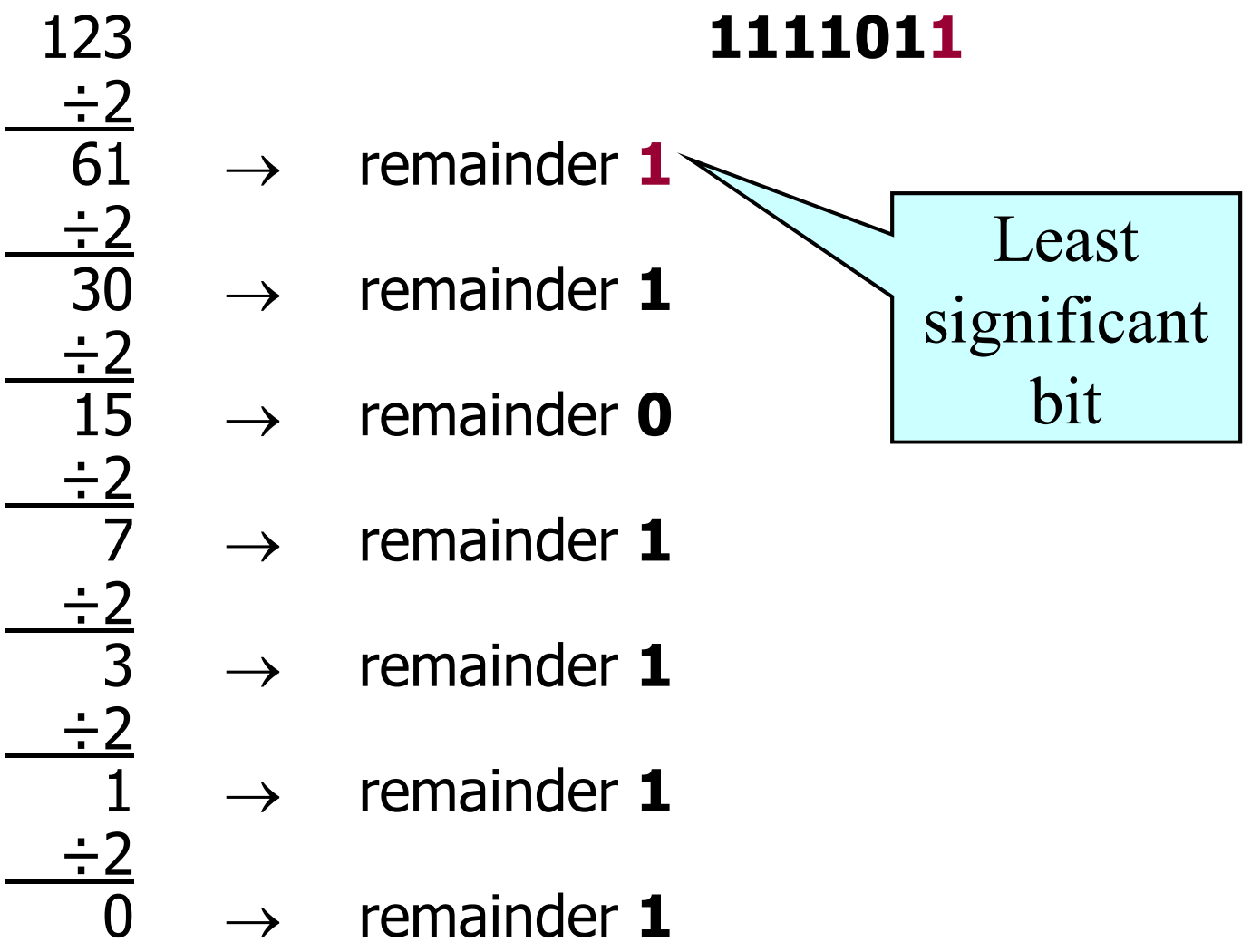
$(123)_{10} = (1111011)_2$

123			**1111011**
÷2			
61	→	remainder **1**	
÷2			
30	→	remainder **1**	
÷2			
15	→	remainder **0**	
÷2			
7	→	remainder **1**	
÷2			
3	→	remainder **1**	
÷2			
1	→	remainder **1**	
÷2			
0	→	remainder **1**	

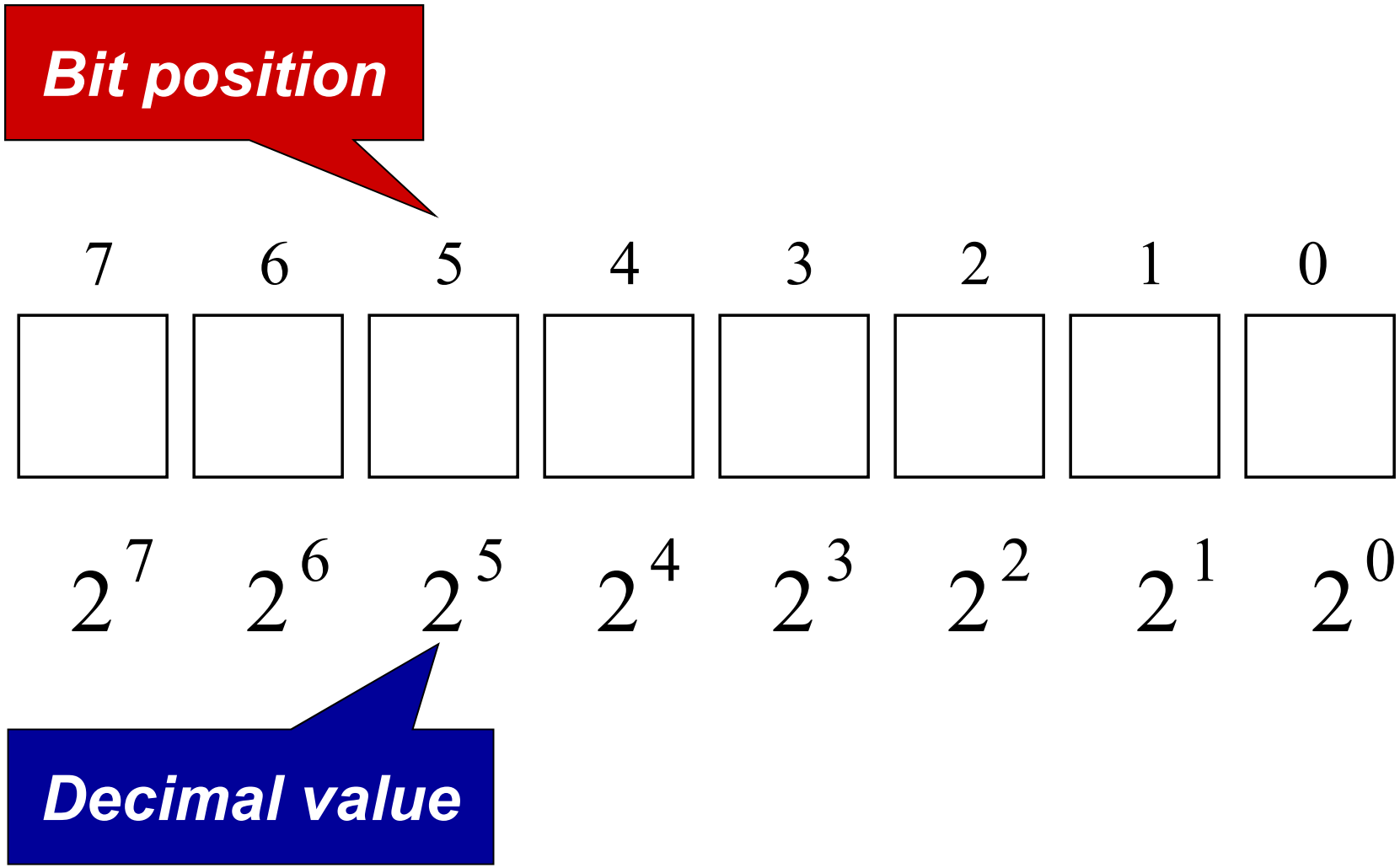
Converting decimal to binary



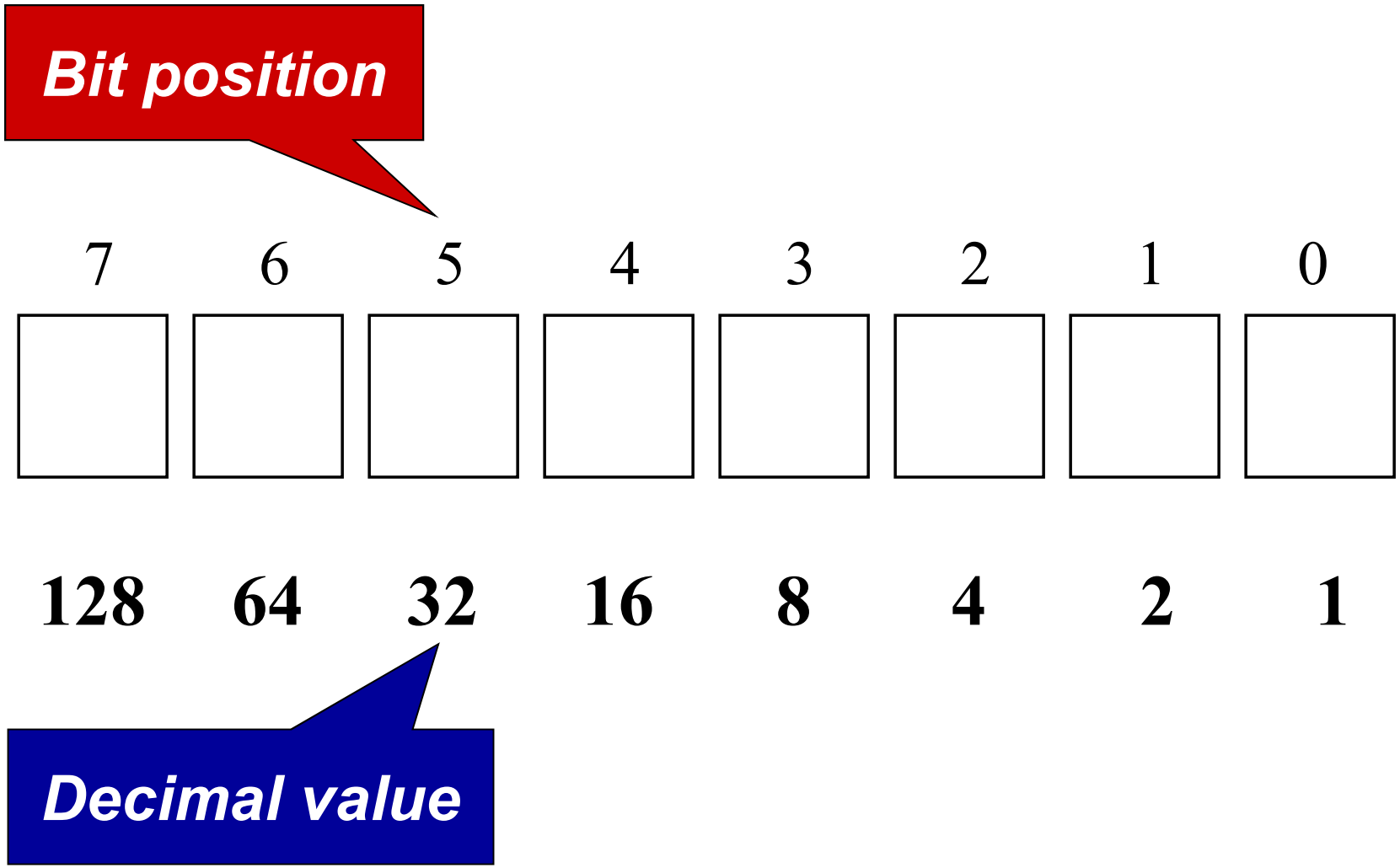
Converting decimal to binary



Converting binary to decimal



Converting binary to decimal



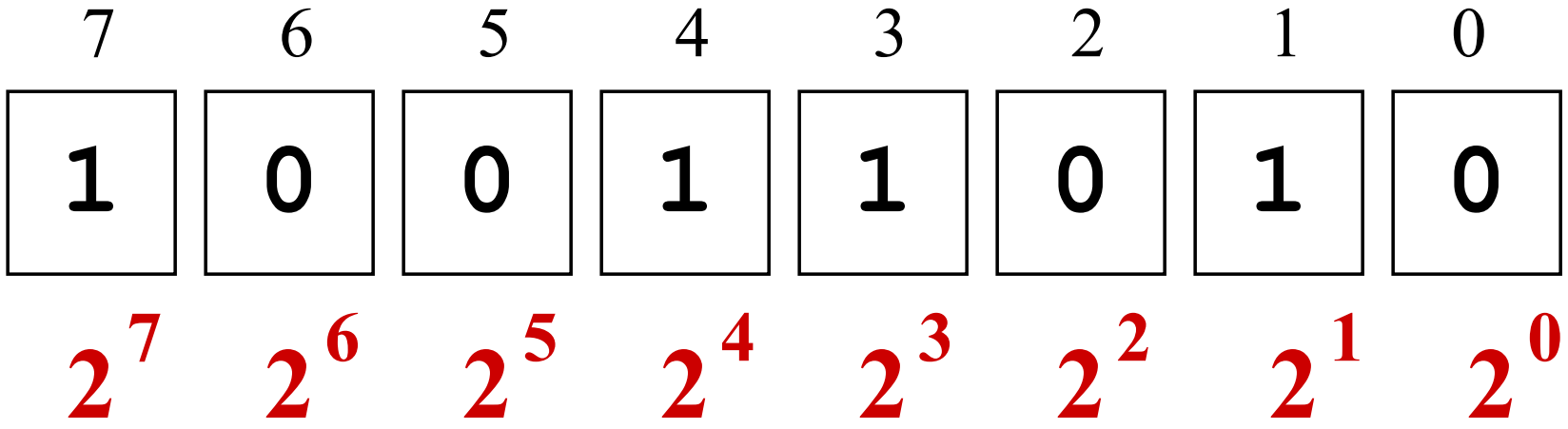
Converting binary to decimal

Example:

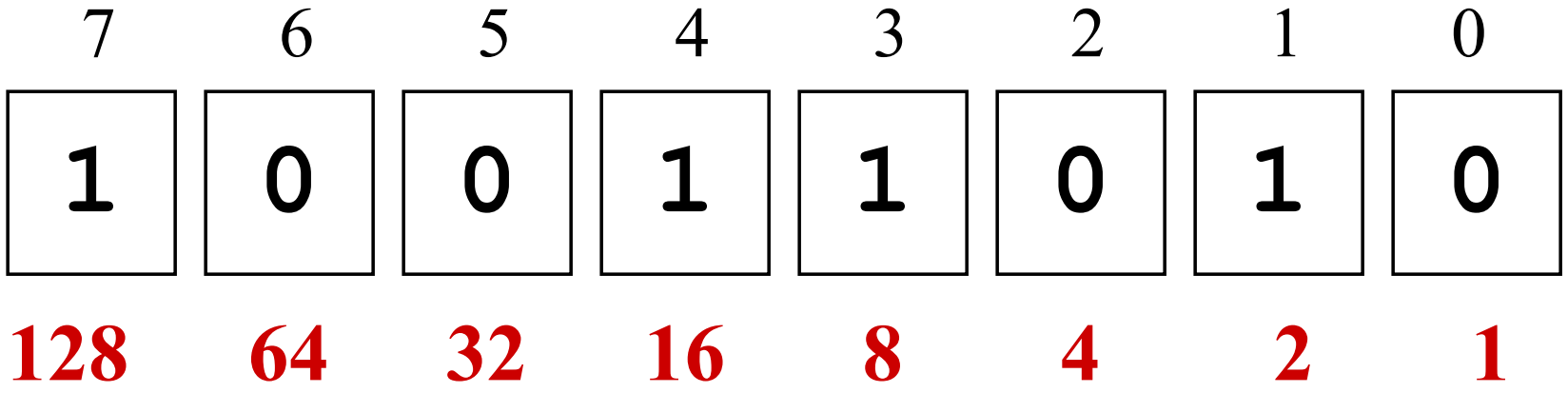
Convert the unsigned binary number `10011010` to decimal

1	0	0	1	1	0	1	0

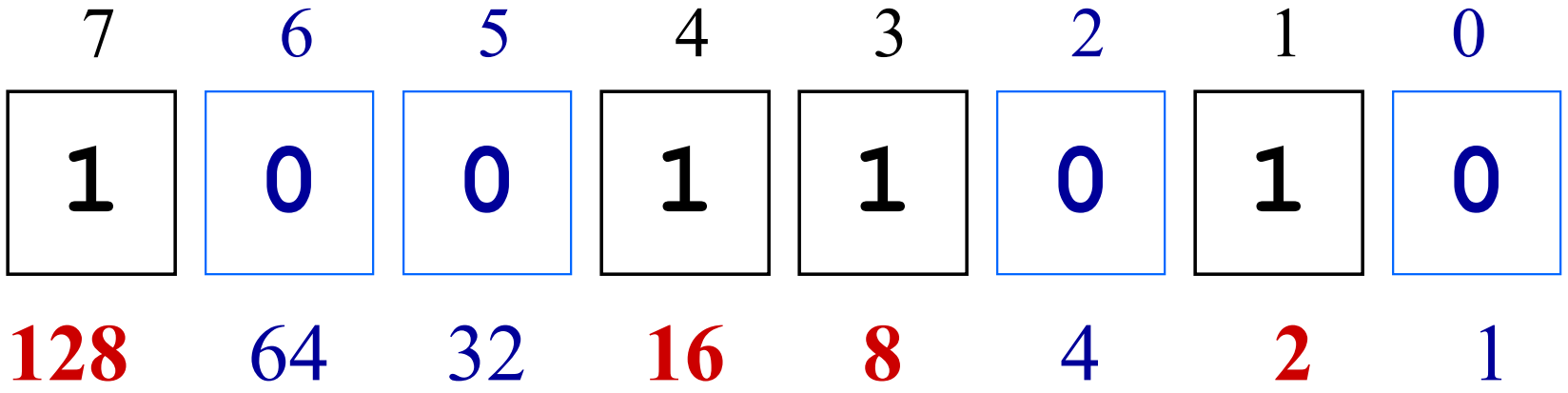
Converting binary to decimal



Converting binary to decimal



Converting binary to decimal



128 + 16 + 8 + 2 = 154

So, `10011010` in **unsigned** binary is **154** in decimal

Converting binary to decimal

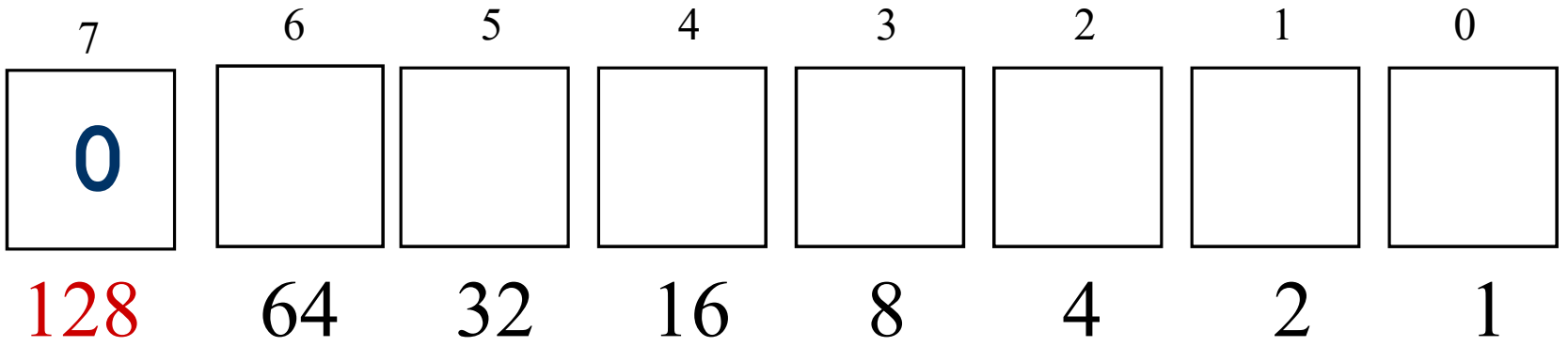
Question:

Convert the decimal number **105** to unsigned binary

Converting binary to decimal

Q. Does 128 fit into **105**?

A. **No**

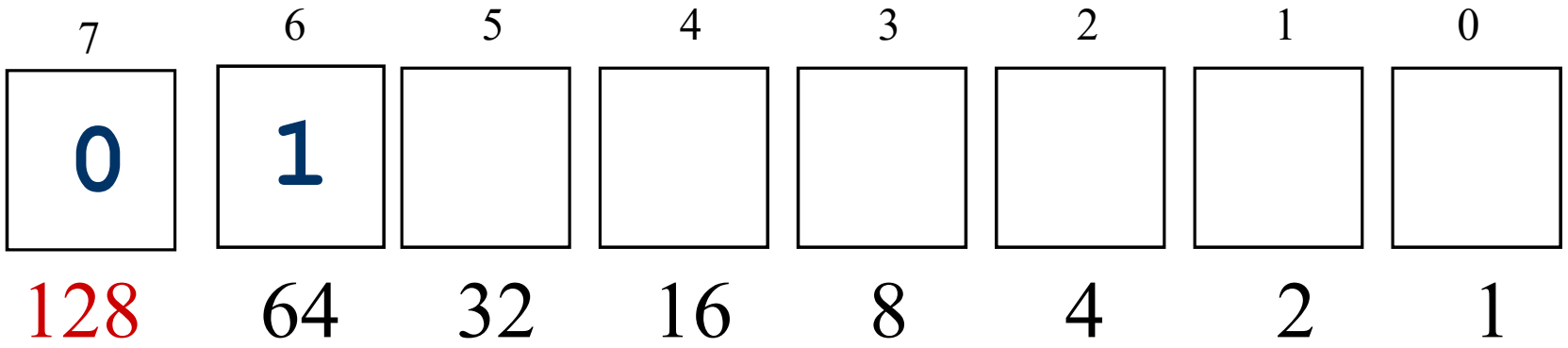


Next, consider the difference: 105- 0*128 = 105

Converting binary to decimal

Q. Does 64 fit into **105**?

A. **Yes**

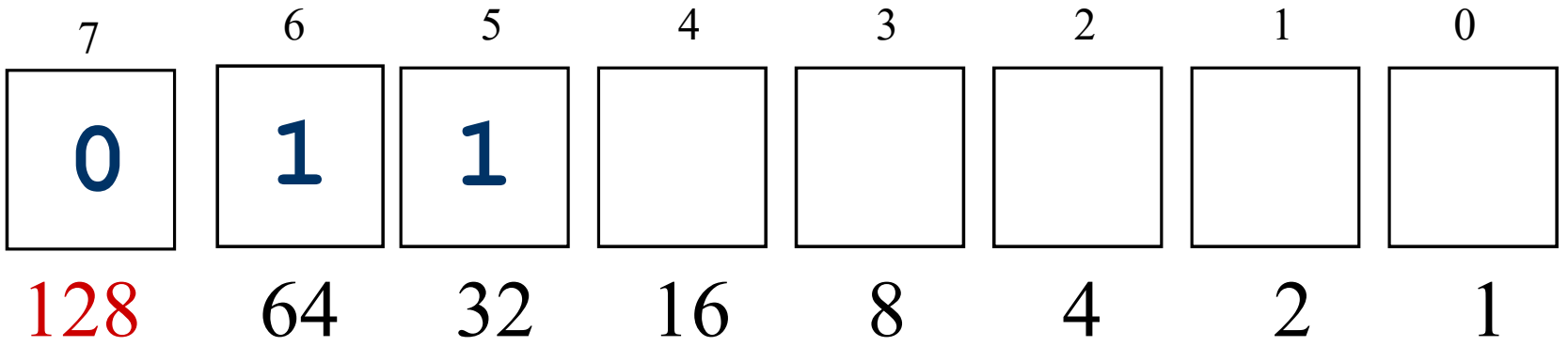


Next, consider the difference: 105- 1*64 = 41

Converting binary to decimal

Q. Does 32 fit into **41**?

A. **Yes**



Next, consider the difference: 41- 32 = 9

Continue your calculation …

Hexadecimal Notation

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

Binary, Decimal & Hexadecimal

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

Binary to Hexadecimal Conversion

10010110_2

1001 **0110**

9 **6**

$10010110_2 = \mathbf{96}$ Hexadecimal

Binary to Hexadecimal Conversion

11011011_2

1101 **1011**

D **B**

$11011011_2 =$ **DB** Hexadecimal

Binary to Hexadecimal Conversion

$00101001\ 11110101_2$

0010	1001	1111	0101
2	9	F	5

$00101001\ 1110101_2 = \mathbf{29F5}$ Hex

ASCII Codes

- American Standard Code for Information Interchange (**ASCII**)
- Use bit patterns of length **seven** to represent
  - Letters of English alphabet: **a - z** and **A - Z**
  - Digits: **0 – 9**
  - Punctuation symbols: **(,), [,], {, }, ’, ”, !, /, **
  - Arithmetic Operation symbols: **+, -, *, <, >, =**
  - Special symbols: **(space), %, $, #, &, @, ^**
- 2^7 = 128 characters can be represented by ASCII, **8th bit** represents the **Parity**.
- Uppercase and Lowercase alphabetic codes differ by 0x**20**.
  - **A** → 0x41 and **a** → 0x61

Character Representation: ASCII

Symbol	ASCII	Symbol	ASCII	Symbol	ASCII	Symbol	ASCII
(space)	00100000	A	01000001	a	01100001	0	00110000
!	00100001	B	01000010	b	01100010	1	00110001
“	00100010	C	01000011	c	01100011	2	00110010
#	00100011	D	01000100	d	01100100	3	00110011
$	00100100	E	01000101	e	01100101	4	00110100
%	00100101	F	01000110	f	01100110	5	00110101
&	00100110	G	01000111	g	01100111	6	00110110
.....		.....		.....		.....	

Character Rep.: ASCII

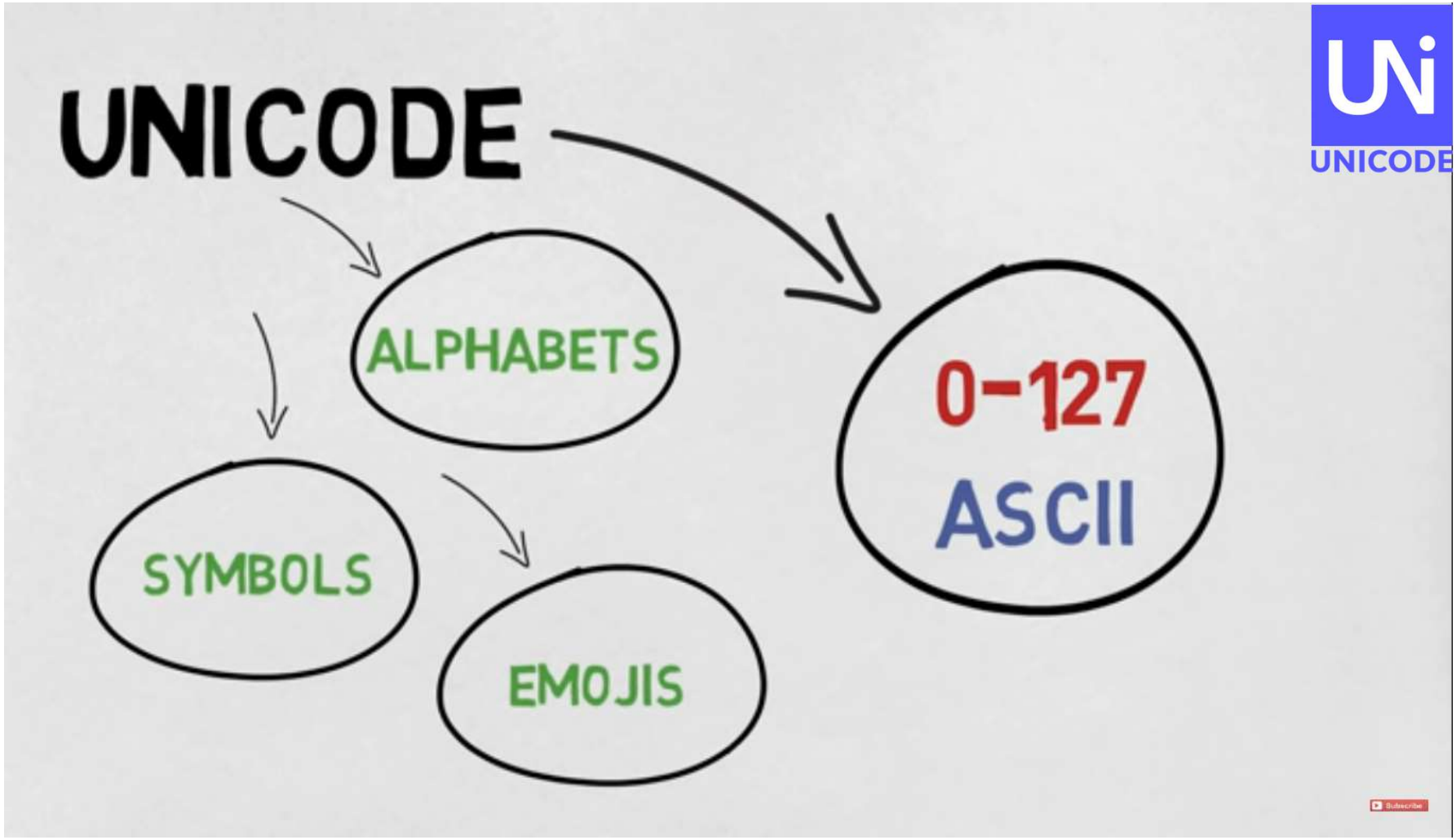
Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Character Rep.: ASCII

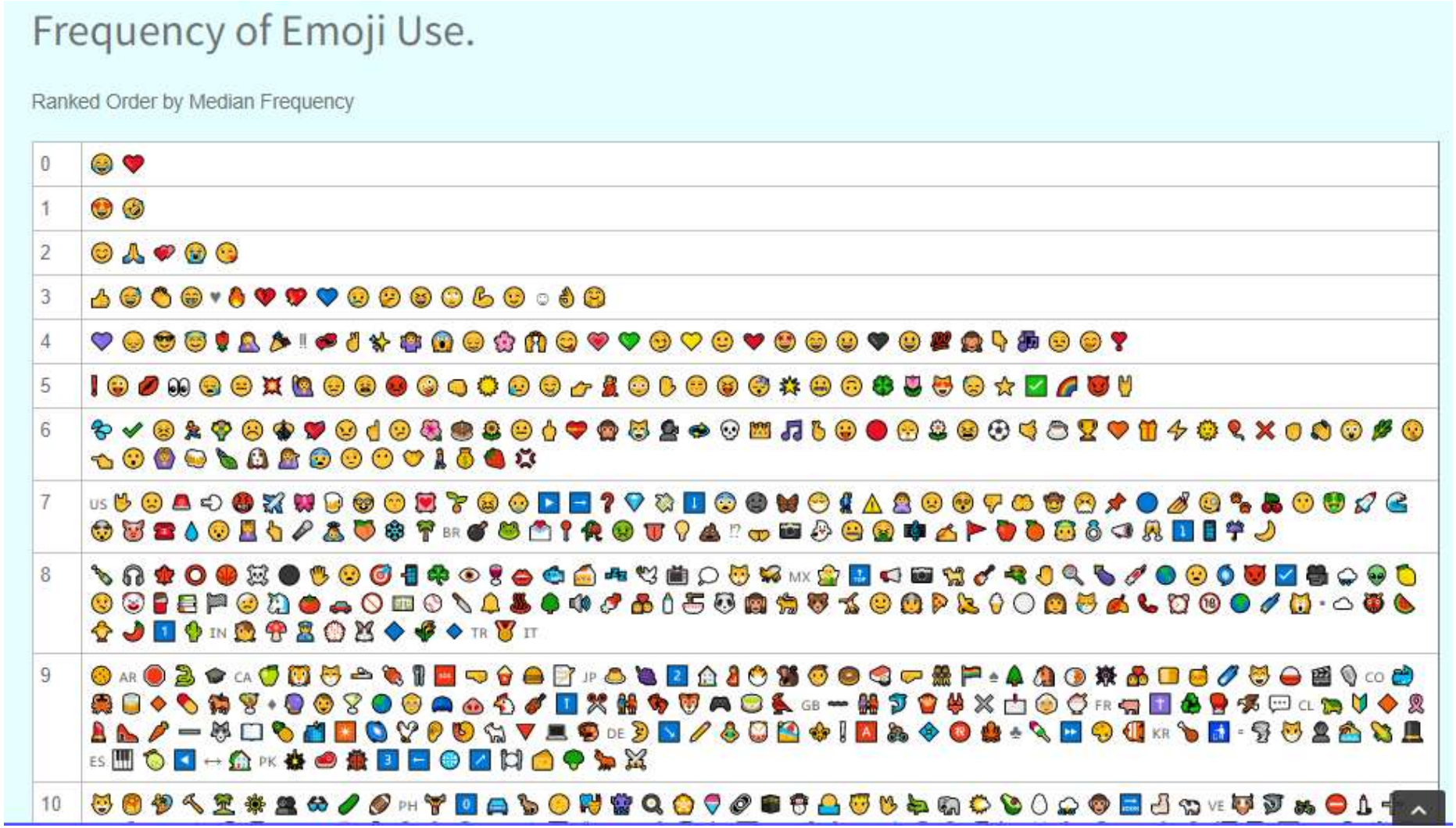
- As computers became more reliable the need for **parity bit** faded.
  - Computer manufacturers **extended ASCII** to provide more characters, e.g., Special characters
  - Used ranges (2^7) =128 ↔ (2^8 - 1) = 255

128	Ç	144	É	161	í	177	▒	193	┴	209	╤	225	ß	241	±
129	ü	145	æ	162	ó	178	▓	194	┬	210	╥	226	Γ	242	≥
130	é	146	Æ	163	ú	179	│	195	├	211	╙	227	π	243	≤
131	â	147	ô	164	ñ	180	┤	196	─	212	╘	228	Σ	244	⌠
132	ä	148	ö	165	Ñ	181	╡	197	┼	213	╒	229	σ	245	⌡
133	à	149	ò	166	ª	182	╢	198	╞	214	╓	230	µ	246	÷
134	å	150	û	167	º	183	╖	199	╟	215	╫	231	τ	247	≈
135	ç	151	ù	168	¿	184	╕	200	╚	216	╪	232	Φ	248	°
136	ê	152	ÿ	169	⌐	185	╣	201	╔	217	┘	233	Θ	249	∙
137	ë	153	Ö	170	¬	186	║	202	╩	218	┌	234	Ω	250	·
138	è	154	Ü	171	½	187	╗	203	╦	219	█	235	δ	251	√
139	ï	156	£	172	¼	188	╝	204	╠	220	▄	236	∞	252	ⁿ
140	î	157	¥	173	¡	189	╜	205	═	221	▌	237	φ	253	²
141	ì	158	₧	174	«	190	╛	206	╬	222	▐	238	ε	254	■
142	Ä	159	ƒ	175	»	191	┐	207	╧	223	▀	239	∩	255	
143	Å	160	á	176	░	192	└	208	╨	224	α	240	≡		

Character Rep.: Unicode



Character Rep.: emoji



https://home.unicode.org/emoji/emoji-frequency/

Character Rep.: Unicode

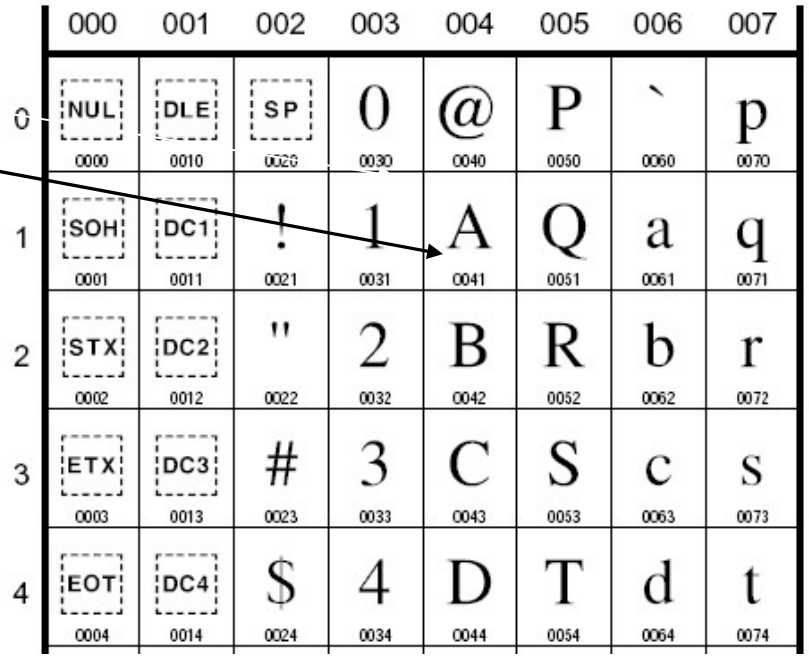
- EBCDIC and ASCII are built around the **Latin alphabet**
  - Are restricted in their ability for representing non-Latin alphabet
  - Countries developed their own codes for native languages
- Unicode: **16-bit system** that can encode the characters of most languages
- 16 bits = 2^{16} = **65,636 characters**
- Most modern programming languages and operating systems now use Unicode as their default character code
- Unicode **code space** is divided into **six parts**
  - The first part is for Western alphabet codes, including English, Greek, and Russian
- **Downward compatible** with ASCII and Latin-1 character sets

Character Rep.: Unicode

- The **Unicode Consortium** was incorporated in California on **3 January 1991**, and in **October 1991**, the **first volume of the Unicode** standard was published.
- Unicode version **3.0** in September **1999** included "**Sinhala**"
- The standard is maintained by the Unicode Consortium, and as of **March 2020**, it has a total of **143,859 characters**, with **Unicode 13.0** (these characters consist of 143,696 graphic characters and 163 format characters) covering 154 modern and historic scripts, as well as multiple symbol sets and emoji.
- The Unicode Standard supports **three** encoding forms (**UTF-8**, **UTF-16**, **UTF-32**) that use a common repertoire of characters.

Character Rep.: Example

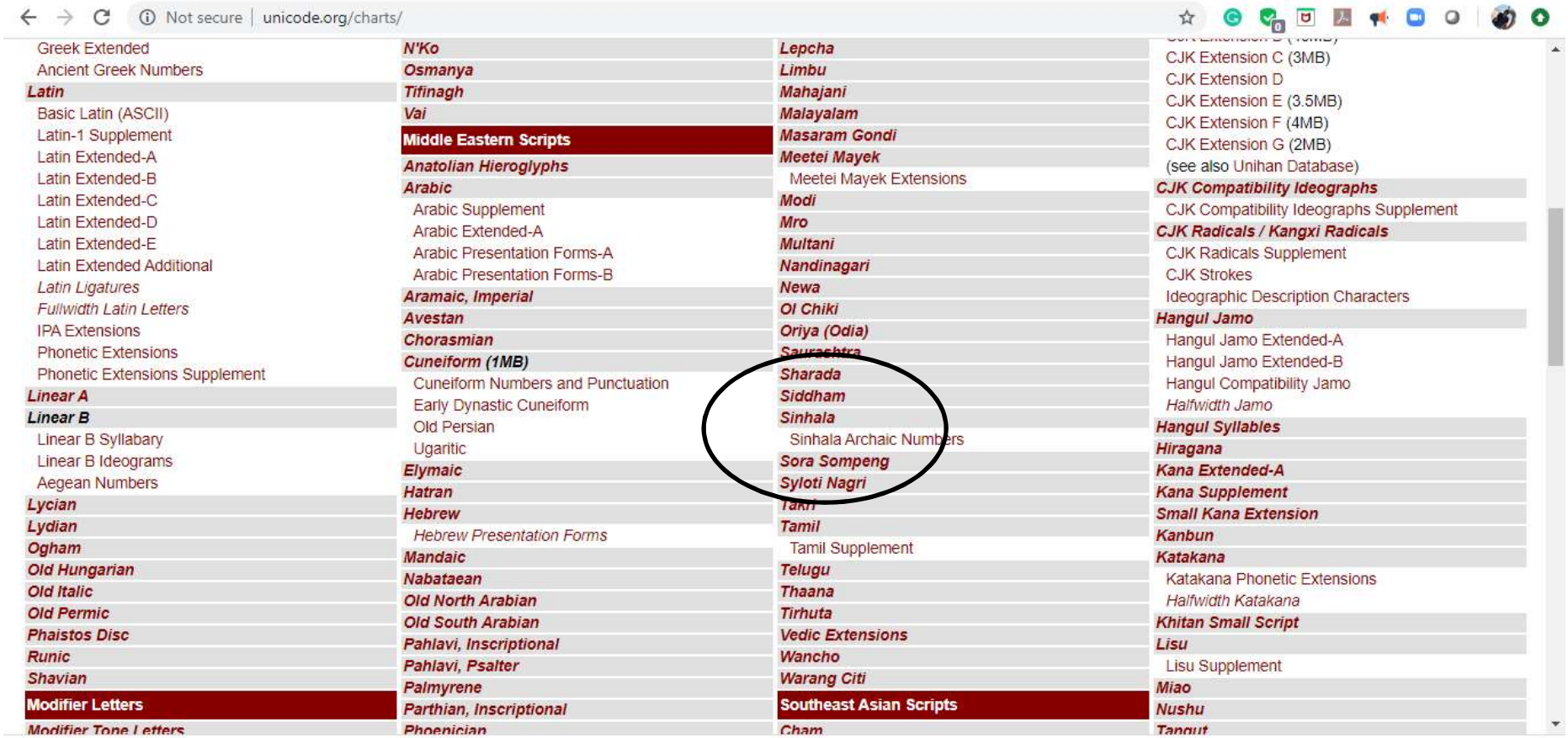
- English section of Unicode Table
  - ACSII equivalent of A is 41_{16}
  - Unicode is equivalent of A:
    - $00\ 41_{16}$



	000	001	002	003	004	005	006	007
0	NUL 0000	DLE 0010	SP 0020	0 0030	@ 0040	P 0050	` 0060	p 0070
1	SOH 0001	DC1 0011	! 0021	1 0031	A 0041	Q 0051	a 0061	q 0071
2	STX 0002	DC2 0012	" 0022	2 0032	B 0042	R 0052	b 0062	r 0072
3	ETX 0003	DC3 0013	# 0023	3 0033	C 0043	S 0053	c 0063	s 0073
4	EOT 0004	DC4 0014	$ 0024	4 0034	D 0044	T 0054	d 0064	t 0074

- Full chart list:
  - http://www.unicode.org/charts/

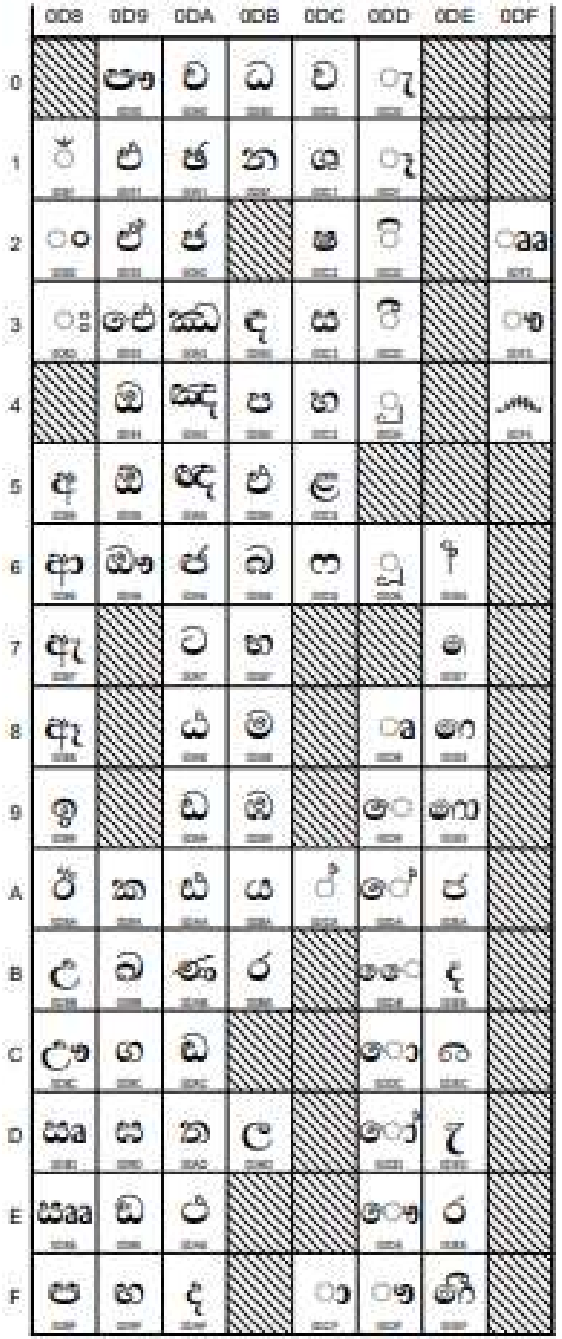
Unicode - Sinhala



Unicode - Sinhala

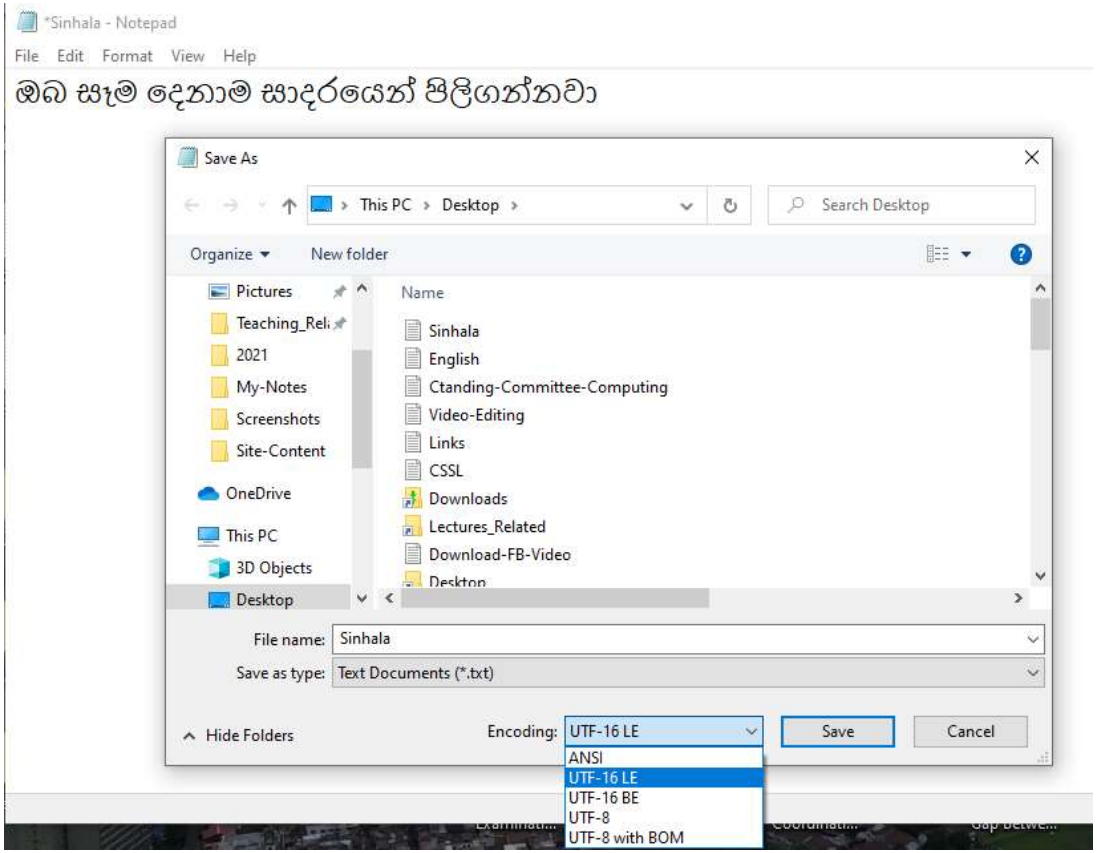
- Characters in Unicode are grouped into **blocks**.
- For example **Sinhala** is in the code page **0D80 to 0DFF** (128 space capacity)
- The Charts do not specify the exact **shape**. They only provide a representative shape for **identification**.

http://www.unicode.org/charts/PDF/U0D80.pdf



Unicode - Sinhala

- Use Notepad application to demonstrate the ASCII and Unicode representation.



UTF-16 (16-bit **Unicode Transformation Format**) is a character encoding, capable of encoding all of Unicode.

Performing Arithmetic

Next Topic …