Lecture 12: Binary Numbers

CS 1109 Summer 2024

Philosophical Questions

- Why is our common number system in base-10?
  - A number system is “base-x” if x numbers can be expressed as a single digit
- Possibly because of how many fingers we have?
- 10 may be an arbitrary choice
  - Why not 5? 12? 26?
- Is it possible that other number systems can be useful?

How does a computer work?

- The basic building block of a modern computer is a transistor
  - For early computers, it was a vacuum tube

How does a computer work?

- The basic building block of a modern computer is a transistor
  - For early computers, it was a vacuum tube
  - Both are functionally similar
- An electrical device with an on and off state
- Why does this matter?

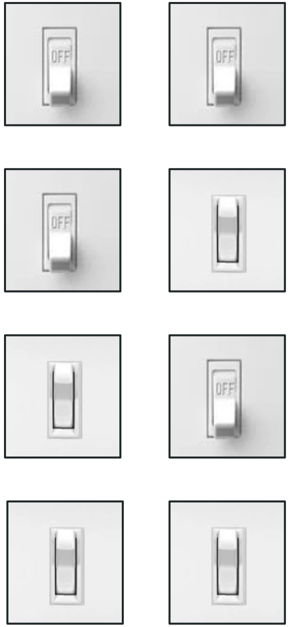
Binary

Binary

- On/off device necessary building block for binary number system
  - Binary is base-2
- Off and on can represent two numbers
- 0 = off and 1 = on

Binary

- If we combine multiple “light switches”, we can create other numbers



= 0

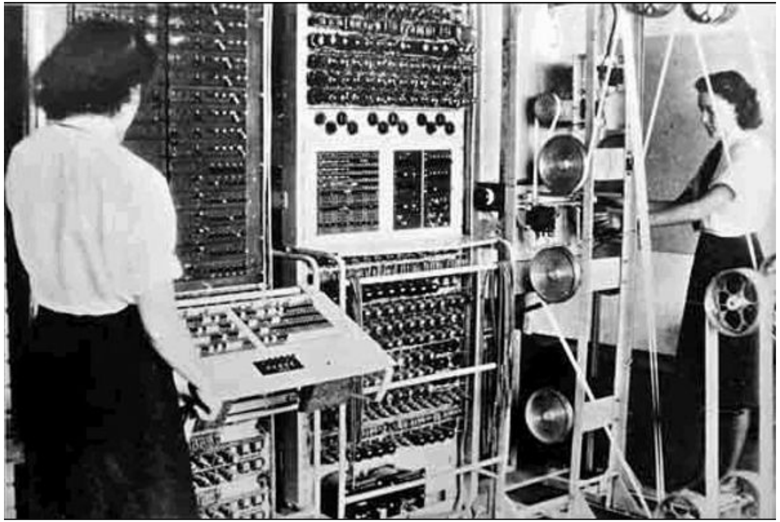
= 1

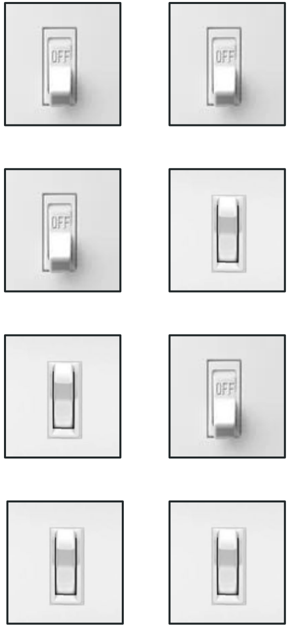
= 2

= 3

Binary

- If we combine multiple “light switches”, we can create other numbers
- Imagine what we could do with thousands of light switches





= 0

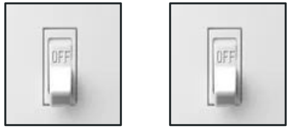
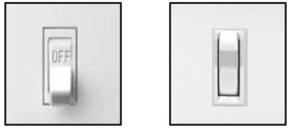
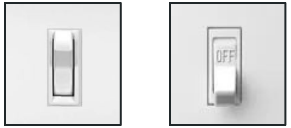
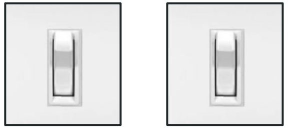
= 1

= 2

= 3

Binary

- Let’s write out the binary for the light switches

Switches	Decimal	Binary
	0	00
	1	01
	2	10
	3	11

Binary

- Bit - portmanteau of “binary digit”
- 1 bit expresses 2 numbers
- 2 bits express 4 numbers
- How many bits to express 8 numbers?
    - 16?

Binary

- Bit - portmanteau of “binary digit”
- 1 bit expresses 2 numbers
- 2 bits express 4 numbers
- How many bits to express 8 numbers?
  - 16?

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

Binary

- Bit - portmanteau of “binary digit”
- 1 bit expresses 2 numbers
- 2 bits express 4 numbers
- How many bits to express 8 numbers?
  - 16?
- Every extra bit doubles how many numbers can be expressed

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

What do digits actually represent?

- Example - 4096 in decimal
- Each digit position represents the base raised to its position
  - Count positions by zero
- Add all the numbers on last row to get number in decimal

Digit	4	0	9	6
Meaning	Thousands	Hundreds	Tens	Ones
Exponent	10^3	10^2	10^1	10^0
Digit * Exponent	4000	000	90	6

Convert from binary to decimal

- Now do the same with binary
- For instance, 1011

Digit	1	0	1	1
Meaning	Eight	Four	Two	One
Exponent	2^3	2^2	2^1	2^0
Digit * Exponent	8	0	2	1

What do digits actually represent?

- Now do the same with binary to convert to decimal
- For instance, 1011
- Add last row - 8 + 2 + 1 = 11
- 1011 in binary is 11 in decimal

Digit	1	0	1	1
Meaning	Eight	Four	Two	One
Exponent	2^3	2^2	2^1	2^0
Digit * Exponent	8	0	2	1

Binary Operations

Basic binary operations

- Addition
  - Example - Add 5 + 9 in binary

Carry-bit				
X				
Y				
X + Y				

Basic binary operations

- Addition
  - Example - Add 5 + 9 in binary

Carry-bit				
X	0	1	0	1
Y	1	0	0	1
X + Y				

Basic binary operations

- Addition
  - Example - Add 5 + 9 in binary

Carry-bit			1	
X	0	1	0	1
Y	1	0	0	1
X + Y				0

Basic binary operations

- Addition
    - Example - Add 5 + 9 in binary

Carry-bit			1	
X	0	1	0	1
Y	1	0	0	1
X + Y			1	0

Basic binary operations

- Addition
  - Example - Add 5 + 9 in binary

Carry-bit			1	
X	0	1	0	1
Y	1	0	0	1
X + Y		1	1	0

Basic binary operations

- Addition
  - Example - Add 5 + 9 in binary

Carry-bit			1	
X	0	1	0	1
Y	1	0	0	1
X + Y	1	1	1	0

Basic binary operations

- Addition
  - Example - Add 5 + 9 in binary
  - 1110 = 14

Carry-bit			1	
X	0	1	0	1
Y	1	0	0	1
X + Y	1	1	1	0

Basic binary operations

- Addition
  - Another example - Add 15 + 1 with a nibble (4 digits)

Carry-bit				
X				
Y				
X + Y				

Basic binary operations

- Addition
  - Another example - Add 15 + 1 with a nibble (4 digits)

Carry-bit				
X	1	1	1	1
Y	0	0	0	1
X + Y				

Basic binary operations

- Addition
  - Another example - Add 15 + 1 with a nibble (4 digits)

Carry-bit			1	
X	1	1	1	1
Y	0	0	0	1
X + Y				0

Basic binary operations

- Addition
  - Another example - Add 15 + 1 with a nibble (4 digits)

Carry-bit		1	1	
X	1	1	1	1
Y	0	0	0	1
X + Y			0	0

Basic binary operations

- Addition
  - Another example - Add 15 + 1 with a nibble (4 digits)

Carry-bit	1	1	1	
X	1	1	1	1
Y	0	0	0	1
X + Y		0	0	0

Basic binary operations

- Addition
  - Another example - Add 15 + 1 with a nibble (4 digits)

Carry-bit (1)	1	1	1	
X	1	1	1	1
Y	0	0	0	1
X + Y	0	0	0	0

Basic binary operations

- Addition
  - Another example - Add 15 + 1 with a nibble (4 digits)
  - 15 + 1 = 0?!!
  - This is integer “overflow”
  - Pay attention to integer sizes in other languages

Carry-bit (1)	1	1	1	
X	1	1	1	1
Y	0	0	0	1
X + Y	0	0	0	0

Basic binary operations

- Addition
- Subtraction (positive numbers only)
    - Example - 11 - 5

X				
Y				
X - Y				

Basic binary operations

- Addition
- Subtraction (positive numbers only)
  - Example - 11 - 5

X	1	0	1	1
Y	0	1	0	1
X - Y				

Basic binary operations

- Addition
- Subtraction (positive numbers only)
  - Example - 11 - 5

X	1	0	1	1
Y	0	1	0	1
X - Y				0

Basic binary operations

- Addition
- Subtraction (positive numbers only)
  - Example - 11 - 5

X	1	0	1	1
Y	0	1	0	1
X - Y			1	0

Basic binary operations

- Addition
- Subtraction (positive numbers only)
  - Example - 11 - 5

X	~~1~~0	10	1	1
Y	0	1	0	1
X - Y		1	1	0

Basic binary operations

- Addition
- Subtraction (positive numbers only)
    - Example - 11 - 5

X	~~1~~0	10	1	1
Y	0	1	0	1
X - Y	0	1	1	0

Basic binary operations

- Addition
- Subtraction (positive numbers only)
  - Example - 11 - 5
  - 0110 = 6

X	~~1~~0	10	1	1
Y	0	1	0	1
X - Y	0	1	1	0

Basic binary logical operations

- Back to truth tables
- AND

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

Basic binary logical operations

- Back to truth tables
- AND
- OR

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

Basic binary logical operations

- Back to truth tables
- AND
- OR
- NOT

A	NOT A
0	1
1	0

More Basics

Binary Sizes

Term	Meaning
Bit	One binary digit
Nibble	4 bits
Byte	8 bits
Kilobyte (KB)*	1024 bits
Megabyte (MB)	1024 KB
Gigabyte (GB)	1024 MB
Terabyte (TB)	1024 GB

* Programmer use of metric units. Some places uses kilo to mean 1000 bits. (see here)

Characters

- In most languages and systems, a character type has size 1 byte (8 bits)
  - Python is different because there is no explicit character type, only strings
- Entire 1972 ASCII table can be expressed with 7 bits

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

Characters

- In most languages and systems, a character type has size 1 byte (8 bits)
  - Python is different because there is no explicit character type, only strings
- Entire 1972 ASCII table can be expressed with 7 bits

Bits				$b_7 b_6 b_5$	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
b_4	b_3	b_2	b_1	Column → / Row ↓	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
1	0	1	1	11	VT	ESC	+	;	K	[	k	{
1	1	0	0	12	FF	FS	,	<	L	\	l	|
1	1	0	1	13	CR	GS	—	=	M	]	m	}
1	1	1	0	14	SO	RS	.	>	N	^	n	~
1	1	1	1	15	SI	US	/	?	O	_	o	DEL

Legibility

- Reading binary numbers gets difficult quickly:
  - 100000000
  - 101011011011
  - 11111111110101
- Need a compressed way to look at binary for legibility
  - Could convert to decimal, but not easy
  - Could we use yet a third number system?

Hexadecimal

- Hexadecimal - base-16 number system
- Every 4 bits is one character from 0-9 and then a-f
- Start with `0x` to show that number is hexadecimal
  - 0x1000
  - 0x40ef
- Used to express:
  - Colors
  - Memory addresses
  - IPv6 network addresses

Number systems table

Binary	Hex	Decimal	Binary	Hex	Decimal
0000	0x0	0	1000	0x8	8
0001	0x1	1	1001	0x9	9
0010	0x2	2	1010	0xb	10
0011	0x3	3	1011	0xa	11
0100	0x4	4	1100	0xc	12
0101	0x5	5	1101	0xd	13
0110	0x6	6	1110	0xe	14
0111	0x7	7	1111	0xf	15

Takeaways

- Number systems are arbitrary
- Decimal used out of necessity (maybe) thanks to fingers
- Binary used out of necessity thanks to transistors
- Hex used out of necessity to help with legibility
- Binary and hex are useful in many aspects of CS
    - Programming - data types and sizes
    - Systems - OS encodings and memory addresses
    - Networking - IPv6 addresses, packet encoding
    - Graphics - color codes