Teaching London Computing

Topic 2.1: Binary Representation

William Marsh
School of Electronic Engineering and Computer Science
Queen Mary University of London

Aims

- Introduce binary number representations
  - Count in binary
  - Do binary arithmetic
- … same rules as normal (decimal), fewer digits

- Understand that binary also represents
  - Characters
  - …

Teaching Issue

- Some people find this material easy …
- … others less so

- Every one can count and do arithmetic!
- How to make it as simple as possible?

At first, avoid mathematics of exponents (2^0, 2^1, 2^2 etc)

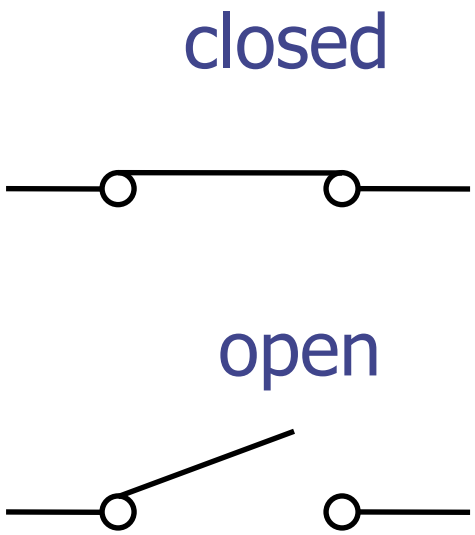
Why Binary?

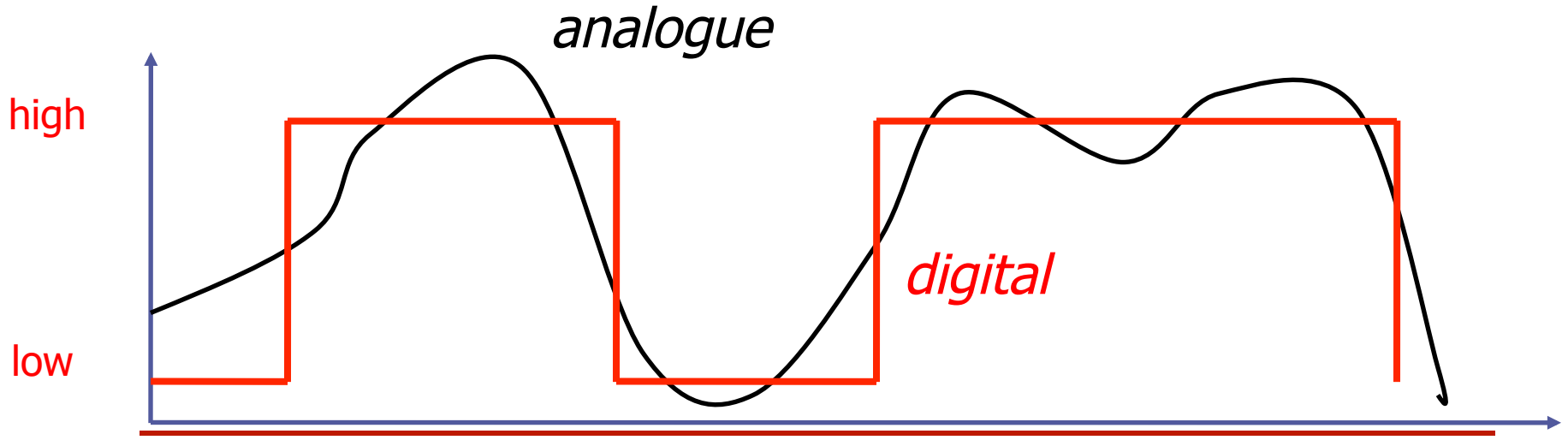
Counting in Tens

- Why do we count with 10 digits (0-9)?
  - Romans used I, II, III, IV, VI, VII …
  - Babylonians counted in 60
  - Our numbers are Arabic – importance of ‘zero’
- Some early computers used decimal
  - Babbage’s mechanical computer
  - ENIAC in USA

Switches make Computers

- Analog versus Digital
  - Analog signal: any value
  - Digital signal: only two values





Counting in Binary

Binary – Base 2

- Decimal
  - Ten digits: ‘0’ to ‘9’
- Two digits: ‘0’ and ‘1’
  - Binary digit – ‘bit’
  - Count on two fingers

Number After – Decimal

- Familiar
- Rules?

0 → 1
1 → 2
9 → 10

18 → 19
19 → 20

99 → 100
100 → 101

999 → 1000

Rule for the next number

Starting from the right hand side, look at each digit 'd' in turn:

1. If d not last → next and stop
2. If d is last → 0, go on to next digit.
3. If there are no more digits, put a '1'

Number After – Binary

- Two digits
- Same rules

```
0 → 1
1 → 10

100 → 101

111 → 1000
1000 → 1001

111111 → 1000000
```

Quiz

- Write down the number after each of the following numbers

```
0 → ??
1 → ??
100 → ??
101 → ??
110 → ??
1000 → ??
1100 → ??
1111 → ??
```

```
01101011 → ??
11000111 → ??
00011010 → ??
10100101 → ??
10111111 → ??
11011111 → ??
01111111 → ??
11111111 → ??
```

Quiz

- Write a list of 4-digit binary numbers in order

0	0000
1	
2	
3	
4	
5	
6	
7	

8	
9	
10	
11	
12	
13	
14	
15	

Binary 0 – 15

- Write a list of 4-digit binary numbers in order

0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

Binary 0 – 15: Patterns

- Write a list of 4-digit binary numbers in order

0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

Binary 0 – 15: Patterns

- Write a list of 4-digit binary numbers in order

0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

Octal and Hexadecimal Numbers

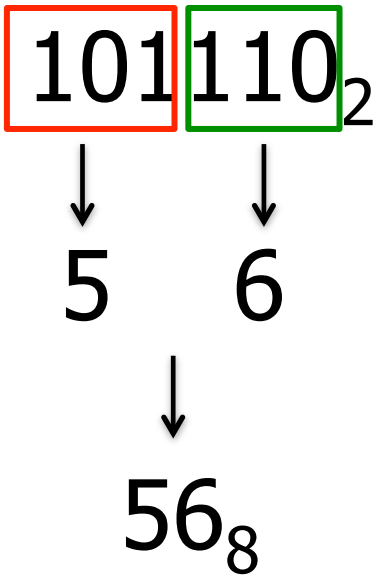
Octal and Hexadecimal

- Octal – base 8
- Hexadecimal – base 16

- Direct conversion to/from binary

Octal – Base 8

- Digits 0 – 7
- Each octal digit is 3 binary digits



octal	binary
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Hex – Base 16

- Digits 0 – 9, A – F
- Each hex digit stands for **4** binary digits

hex	binary	hex	binary
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

Hex – Example

- AC_H

↓ ↓

1010 1100

hex	binary	hex	binary
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

Representing Characters

ASCII

- American Standard Code for Information Interchange
  - First standardised in 1967
- 7-bit character code
  - 0 – 127
  - Parity bit
- Characters are
  - Printable – 95
  - Control – 33

```
!"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_
`abcdefghijklmnopqrstuvwxyz{|}~
```

ASCII in Hex

	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	SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
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

- 'A' is $41_H = 65$
- '2' is $32_H = 50$

ASCII Features – Digits

- The digits 0-9 are represented with their values in binary prefixed with 0011 (I.e. 3_H)
  - '0' is 30_H
  - '9' is 39_H

ASCII Features – Characters

- Letters are in a contiguous sequences
  - No gaps
- Lowercase and uppercase letters 32_{10} apart
  - Single bit difference
  - A – Z : 41_H – $5A_H$
  - a – z : 61_H – $7A_H$

ASCII Limitations

- Only 95 printable characters
  - accents
  - non-Latin languages
  - Chinese-based scripts
- Unicode
  - 16 bits
  - Overlaps with ASCII where possible

Adding Binary Numbers

Adding Binary Digits

- Very simple rules

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0 carry 1

Adding Binary Numbers

- Complete the following

```
    0 1 1 0
+   0 1 0 1
  ___________
```

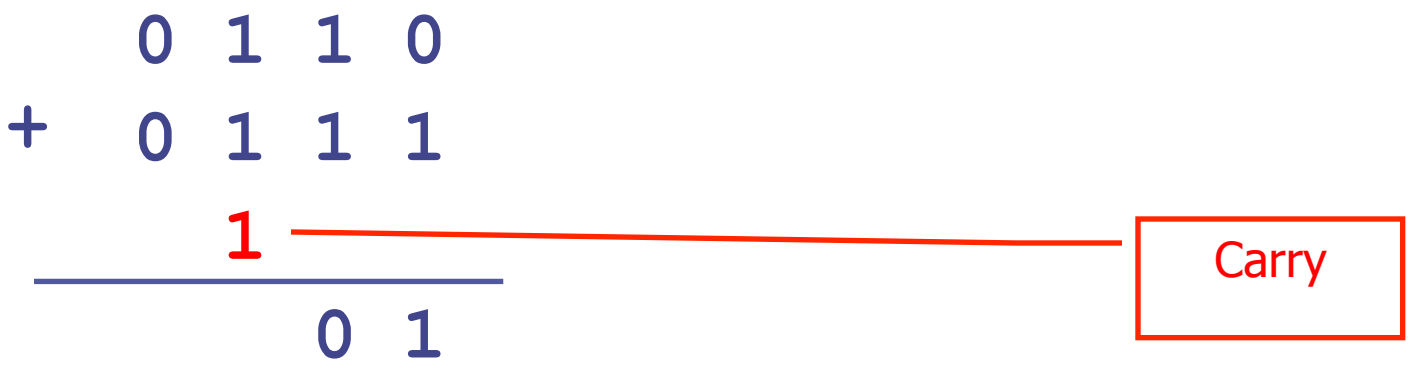
Adding Binary Numbers

- Complete the following

```
    0 1 1 0
+   0 1 1 1
-------------
          1
```

Adding Binary Numbers

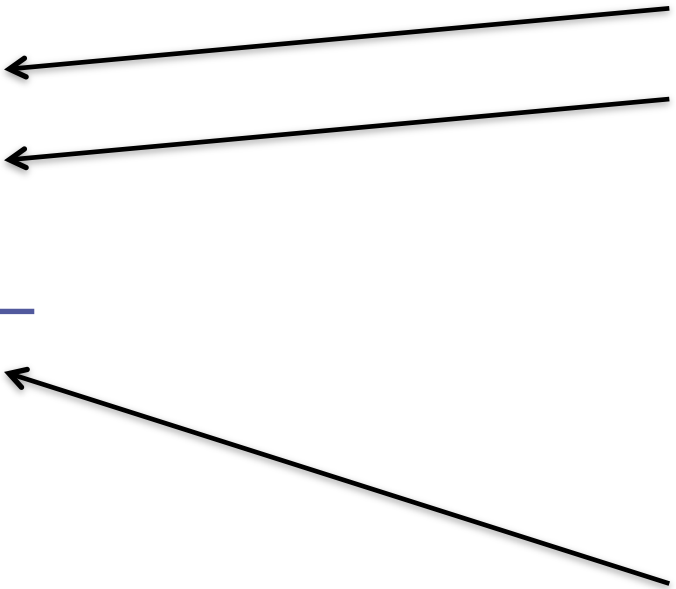
- Complete the following



Adding Binary Numbers

- Complete the following

```
   0 1 1 0
+  0 1 1 1
   1 1
  ---------
   1 1 0 1
```



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

Quiz

- Complete

```
    0 1 1 1 0 1
+   0 0 0 1 0 1
  ---------------
```

Summary

- Start by counting … get used to binary
  - Conversion tables
- Addition – apply standard rules
- Octal and hexadecimal
  - Easy really

At first, avoid mathematics of exponents (2^0, 2^1, 2^2 etc)