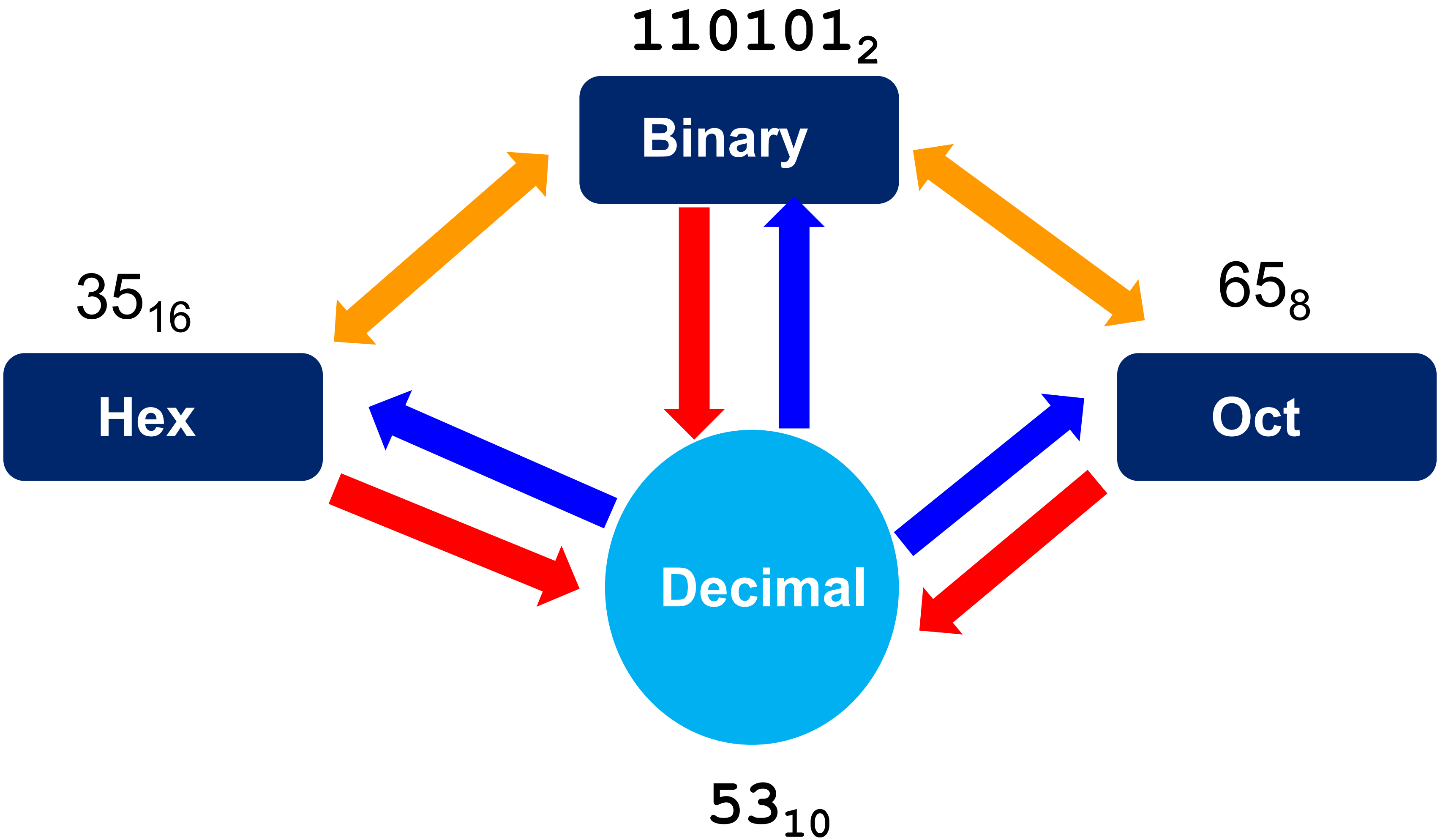
Queen Mary
University of London
Science and Engineering

EBU4202: Digital Circuit Design
Number Systems & Codes Tutorials

Dr. Md Hasanuzzaman Sagor (Hasan)
Dr. Chao Shu (Chao)
Dr. Farha Lakhani (Farha)

School of Electronic Engineering and Computer Science,
Queen Mary University of London,
London, United Kingdom.

Conversion



Digits in base **R** range from **0** to **(R – 1)**.

Queen Mary
University of London

Conversion: Any Base to Decimal

- **Example**: Conversion of binary to decimal

  $10011_2 = ?$

- **Example**: Conversion of base 5 to decimal

  $4321_5 = ?$

- **Example**: Conversion of base 5 to decimal

  $12.3_5 = ?$

Queen Mary
University of London

Non-integer to Binary Conversion

- **Example**: Convert 225.50 to binary

- **Example**: 110110111001_2 to decimal

- **Example**: Conversion of binary to octal

$$101110_2 = (?)_8$$

Review: Subtraction with 2's Complement

This can be rewritten as: $23_{10} + (-45_{10})$, so we need to convert the binary representation of 45_{10} to two's complement.

```
45₁₀  =  0010 1101
         1101 0010     ← Invert
        +0000 0001     ← Add 1
         1101 0011 = -45₁₀
```

Now just add!

```
    23      0001 0111
+  -45     +1101 0011
   -22      1110 1010
           -0000 0001   ← Subtract 1
            1110 1001
            0001 0110   ← Invert to convert to simple binary
```

Check

This is -22_{10} really, as we're just checking!

$= 1*2^4 + 0*2^3 + 1*2^2 + 1*2^1 + 0*2^0$

$= 16 + 0 + 4 + 2 + 0 = 22_{10}$

To subtract two signed numbers, take the 2's complement of the subtrahend and add. Discard any final carry bit.

Queen Mary University of London

Binary – 2's Complement

- Conversion to 2's complement:
  - *Positive numbers*: same as simple binary.
  - *Negative numbers*:
    1. Obtain the *n-bit* simple binary equivalent.
    2. Invert the bits of that representation.
    3. Add *1* to the result.

Example 1: Convert -276_{16} to 16-bit 2's complement

Example 2: Perform the following subtraction of signed numbers: $15_{10} - 6_{10}$

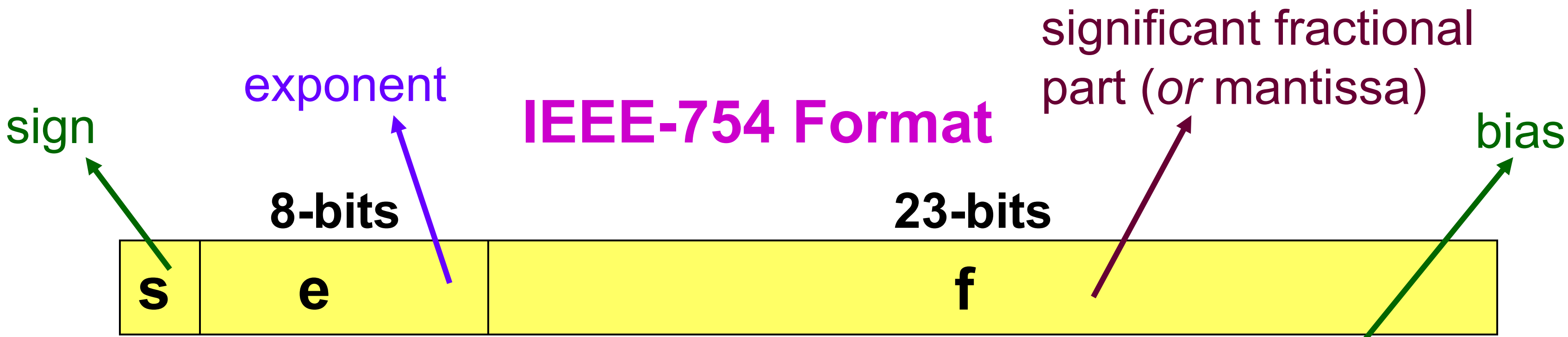
Example 3: Perform the following subtraction of signed numbers: 00001100 - 11110111 $[12_{10} - (-9_{10})]$

Example 4: Perform the following subtraction of signed numbers: 11100111 – 00010011 $[-25_{10} - (+19_{10})]$

Queen Mary
University of London

Review: Floating Point Formats (2/3)

IEEE-754 Format



Normal Value = $(-1)^s\ 1.f * 2^{\,e - 127}$

Denormal Value = $(-1)^s\ 0.f * 2^{\,1 - 127}$

Denormals are used for values very close to zero.

Example: IEEE-754 FP

1. Represent 145.84375_{10} in floating point format:

2. Convert the IEEE-754 floating point number

x = 01000011010111010000000000000000 [16 zeros] to a decimal number:

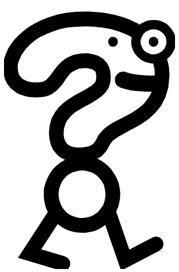
Queen Mary
University of London

Review: Odd and Even Parity

What is actually sent when even parity has been agreed.

	'S'	'E'
ASCII	101 0011	100 0101
Even parity	0101 0011	1100 0101
Odd Parity	1101 0011	0100 0101

- **Example** (detection of a 1-bit error):
  - ASCII '**S**' is sent (i.e., send **1010011_2**), but value **01010010_2** is received.



Assuming value 01010010_2 is received, can Parity Checking detect an error when character 'S' is sent, if *even parity* has been agreed?

sent = 01010011 ('S')
received = 01010010
Received value has odd number of 1's (so doesn't obey even parity), so error is detected (1 bit flipped).

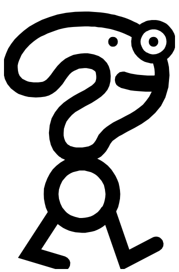
Queen Mary
University of London

Example: Odd and Even Parity

What is actually sent when even parity has been agreed.

	'S'	'E'
ASCII	101 0011	100 0101
Even parity	0101 0011	1100 0101
Odd Parity	1101 0011	0100 0101

- **Example** (detection of a 1-bit error):
  - ASCII '**E**' is sent (i.e., send **1000101_2**), but value **01010010_2** is received.



What if instead we send character 'E' and *odd parity* has been agreed? Can Parity Checking detect an error?

Queen Mary University of London

Example: Odd and Even Parity

What is actually sent when even parity has been agreed.

	'S'	'E'
ASCII	101 0011	100 0101
Even parity	0101 0011	1100 0101
Odd Parity	1101 0011	0100 0101

- **Example:** (detection of a 1-bit error):
  - ASCII '**E**' is sent (i.e., send **1000101_2**) and even parity has been agreed, but value **00000101_2** is received. Can Parity Checking detect an error?

Queen Mary University of London

Queen Mary
University of London
Science and Engineering

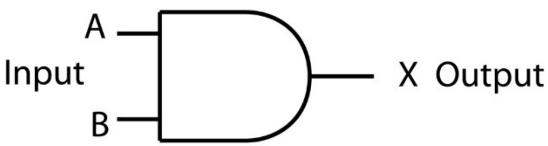
EBU4202: Digital Circuit Design
Switching Algebra Tutorial

Dr. Md Hasanuzzaman Sagor (Hasan)
Dr. Chao Shu (Chao)
Dr. Farha Lakhani (Farha)

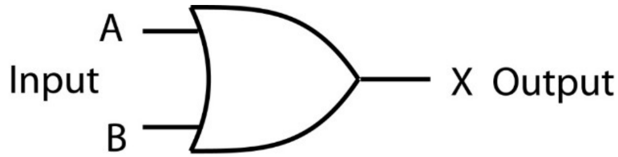
School of Electronic Engineering and Computer Science,
Queen Mary University of London,
London, United Kingdom.

Review Gates

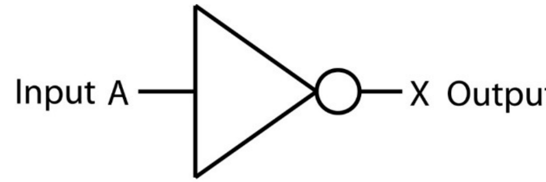
AND



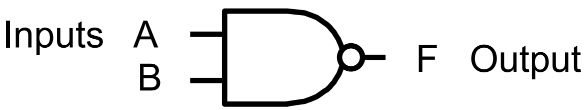
OR



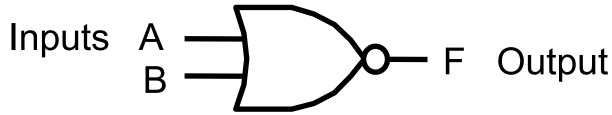
NOT



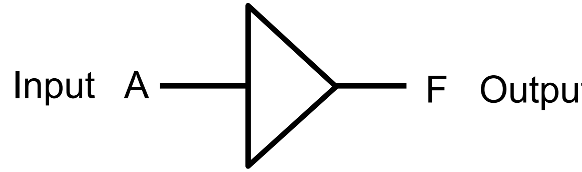
NAND



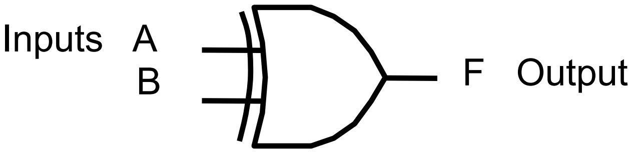
NOR



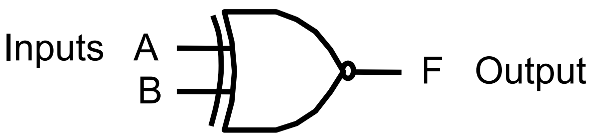
Non-inverting buffer



XOR

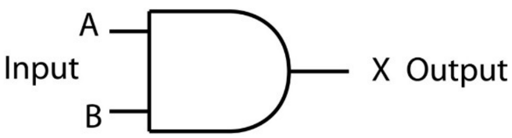


XNOR



Review Gates

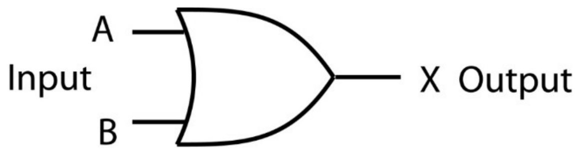
AND



Input A	Input B	Output X
0	0	0
0	1	0
1	0	0
1	1	1

$X = A.B$

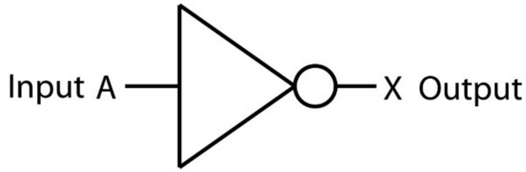
OR



Input A	Input B	Output X
0	0	0
0	1	1
1	0	1
1	1	1

$X = A+B$

NOT



Input A	Output X
0	1
1	0

$X = A'$

Queen Mary
University of London

Review Gates

NAND

A	B	F
0	0	1
0	1	1
1	0	1
1	1	0

$F = (A.B)'$

NOR

A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

$F = (A+B)'$

XOR

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

$F = (A \oplus B)$

$F = A'B+AB'$

XNOR

A	B	F
0	0	1
0	1	0
1	0	0
1	1	1

$F = (A \oplus B)'$

$F = A'B'+AB$

Adsorption Theorem

Adsorption Theorem (not in the textbook):
(T*) $X + X'Y = X + Y$ **(T*)'** $X(X' + Y) = XY$

Proof:

$$X + X'Y = (X + X')(X + Y) \quad (T8')$$
$$= 1.(X+Y) \quad (T5)$$
$$= X+Y \quad (T1')$$

(T8') $(X + Y)(X + Z) = X + YZ$

(T5) $X+X' = 1$

(T1') $X \cdot 1 = X$

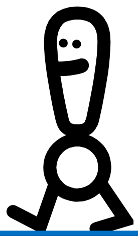
Queen Mary
University of London

Example (1/3): Two Equations

- Show that $F_1 = F_2$ using a Truth Table.

$$F_1 = X' \bullet Y' \bullet Z + X' \bullet Y \bullet Z + X \bullet Y'$$

$$F_2 = X' \bullet Z + X \bullet Y'$$



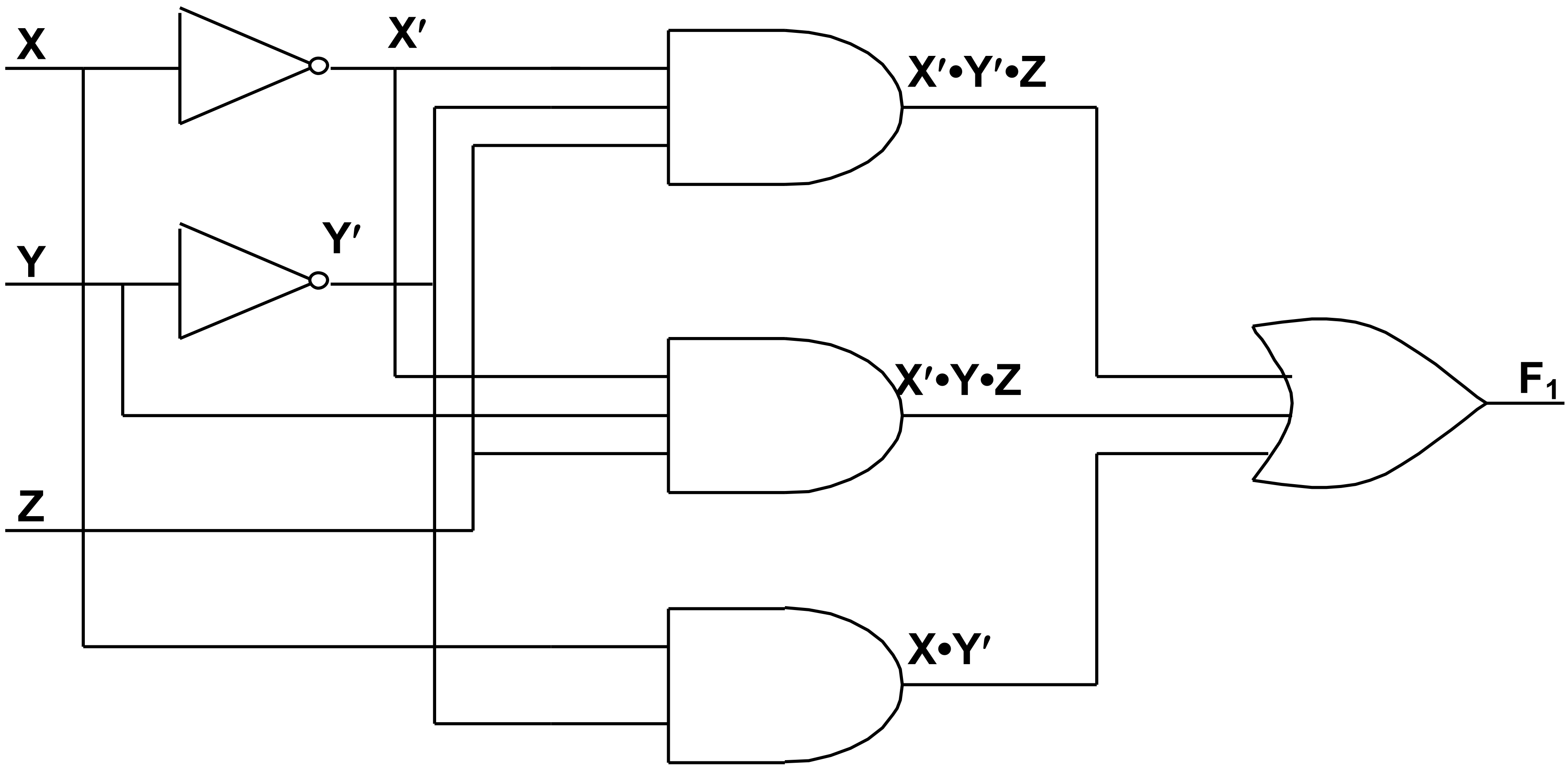
To prove the equality, we can use Switching Algebra theorems **T8** + **T5**.

(T8) → $XY + XZ = X(Y + Z)$
(T5) → $X + X' = 1$
Show: prove that $F_1 = F_2$ using Switching Algebra,
$F_1 = X'Y'Z + X'YZ + XY' =$
$= X'Z(Y' + Y) + XY'$, using (T8)
$= X'Z + XY'$, using (T5)
$= F_2$

X	Y	Z	F_1	F_2
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	0	0
1	1	1	0	0

Example (2/3): Two Equations

$F_1 = X'\bullet Y'\bullet Z + X'\bullet Y\bullet Z + X\bullet Y'$



Example (3/3): Two Equations

To be completed in class …

$F_2 = X' \bullet Z + X \bullet Y'$

Queen Mary
University of London

Example 2: Minimisation (1)

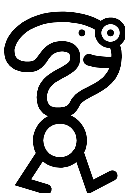
$F = A'B'C' + A'B'C + A'BC$

$A'B'C = A'B'C + A'B'C$ ***(T3) X + X = X***

$$F = \underbrace{A'B'C' + A'B'C}_{\textbf{T10}} + \underbrace{A'B'C + A'BC}_{\textbf{T10}}$$

(T10) XY + XY' = X

$F = A'B' + A'C$



Circuit Diagrams for these functions?

Note: Can be further simplified to F = A'(B'+C)

Example 2: Minimisation (2)

To be completed in class …

$F = A'B'C' + A'B'C + A'BC$

$F = A'B' + A'C$

Exercise

Simplify the following two functions:

a) G=(A+B)(A+C')(A+D)(BC'D+E)

Queen Mary
University of London