Precept 3

Main topic:
- Cyclic redundancy check (CRC)

Breakout rooms:
- Reminder: participation is part of your grade
- What types of errors can be detected?
- Is there a pattern in how errors are detected?

Cyclic redundancy check (CRC)

- Most popular method **error detecting code** at L2
  - Found in Ethernet, Wi-Fi, token ring, many many others

- Often implemented in hardware at the link layer

- Represent ***k*-bit messages** as **degree *k* − 1 polynomials**
  - **Each coefficient in the polynomial is either zero or one, *e.g.*:**

$$\overbrace{\boxed{1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0}}^{k\,=\,6 \text{ bits of message}}$$

$$M(x) = 1x^5 + 0x^4 + 1x^3 + 1x^2 + 1x + 0$$

Modulo-2 Arithmetic

- **Addition** and **subtraction** are **both** **exclusive-or without carry or borrow**

Multiplication example:

```
  1101
   110
------
  0000
 11010
110100
------
101110
```

Division example:

```
       1101
    ______
110 |101110
     110
     ---
      111
      110
      ---
       011
       000
       ---
        110
        110
        ---
```

CRC at the sender

- ***M*(x)** is our **message** of length ***k***
  - **e.g.:** $M(x) = x^5 + x^3 + x^2 + x \quad (k = 6)$

1 0 1 1
1 0

- Sender and receiver agree on a ***generator*** polynomial ***G*(*x*)** of degree ***g* − 1 (*i.e.*, *g* bits)**
  - **e.g.:** $G(x) = x^3 + 1 \quad (g = 4)$

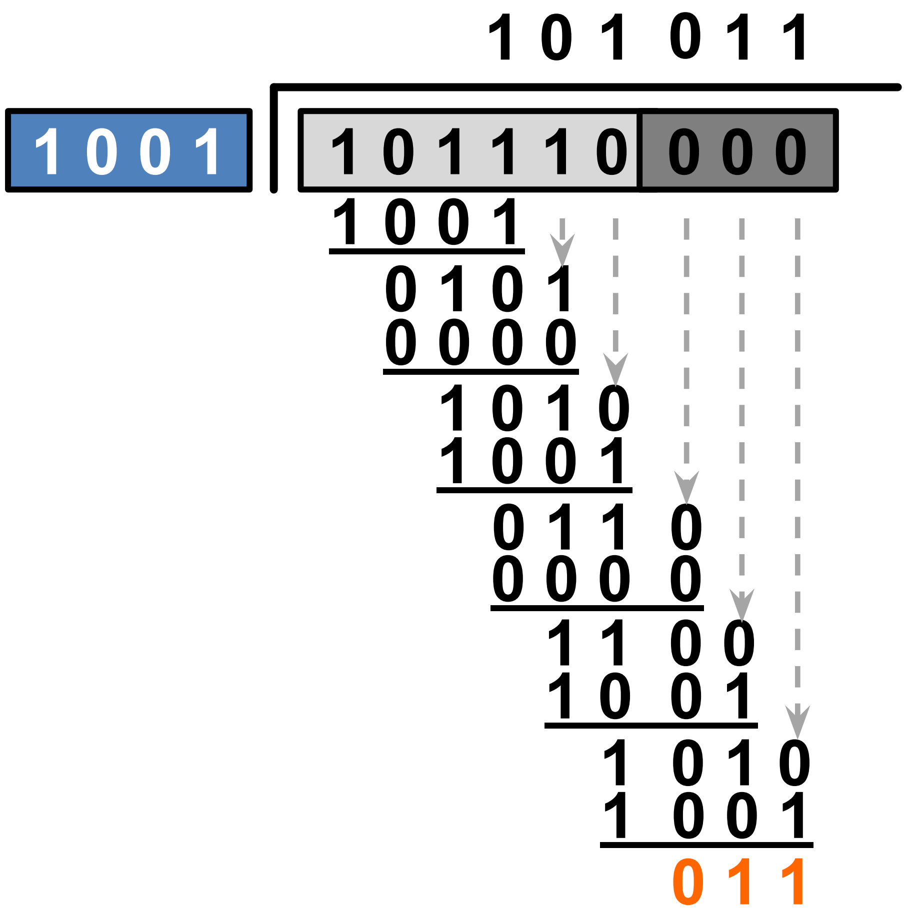
1

1. Calculate **padded message $T(x) = M(x) \cdot x^{g-1}$**
   - *i.e.*, right-pad with $g - 1$ zeroes
   - **e.g.:** $T(x) = \mathrm{M(x)} \cdot x^3 = x^8 + x^6 + x^5 + x^4$

1 0 1 1 0 0
1 0 0

CRC at the sender

2. **Divide** padded message ***T*(*x*)** by generator ***G*(*x*)**
   – **The remainder *R*(*x*)** is the **CRC:**



***R*(*x*) = *x* + 1**

CRC at the sender

3. **The sender transmits codeword $C(x) = T(x) + R(x)$**
   - *i.e.*, the sender transmits the original message with the CRC bits appended to the end

   - Continuing our example, $C(x) = x^8 + x^6 + x^5 + x^4 + x + 1$

`1 0 1 1 0 1`

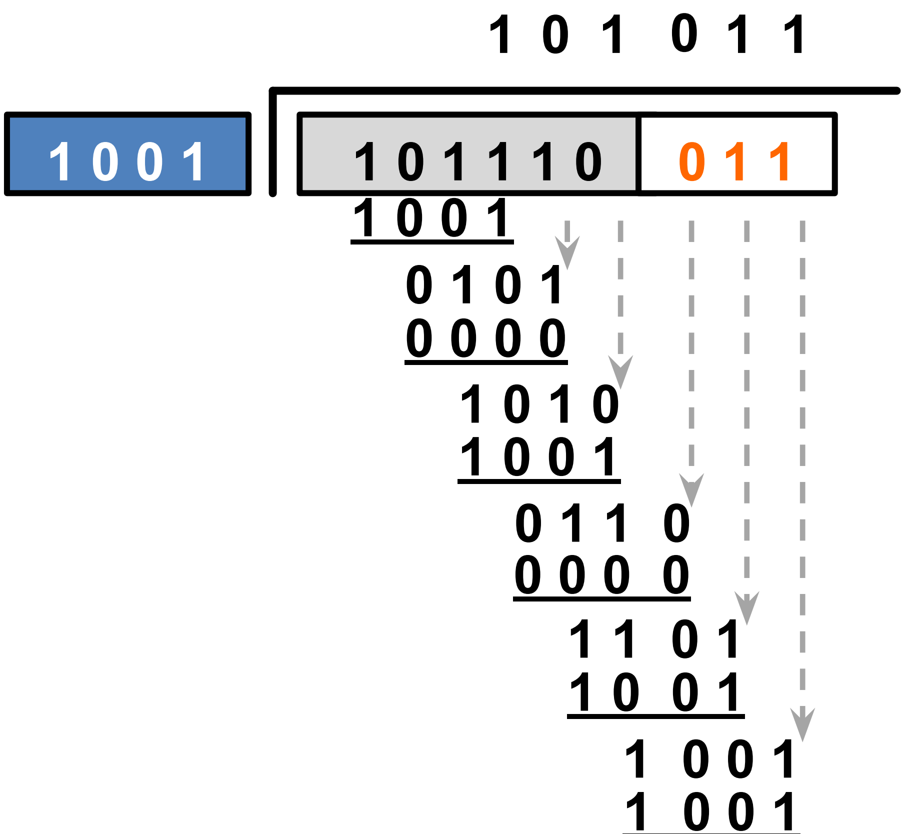
1 0	1

Properties of CRC codewords

- Remember: **Remainder** [$T(x)/G(x)$] = $R(x)$
- What happens when we divide $C(x)$ / $G(x)$?
- $C(x) = T(x) + R(x)$ so **remainder** is
  - **Remainder** [$T(x)/G(x)$] = $R(x)$, **plus**
  - **Remainder** [$R(x)/G(x)$] = $R(x)$
  - Recall, **addition is exclusive-or operation**, so:
    - **Remainder** [$C(x)/G(x)$] = $R(x) + R(x) = 0$

Detecting errors at the receiver

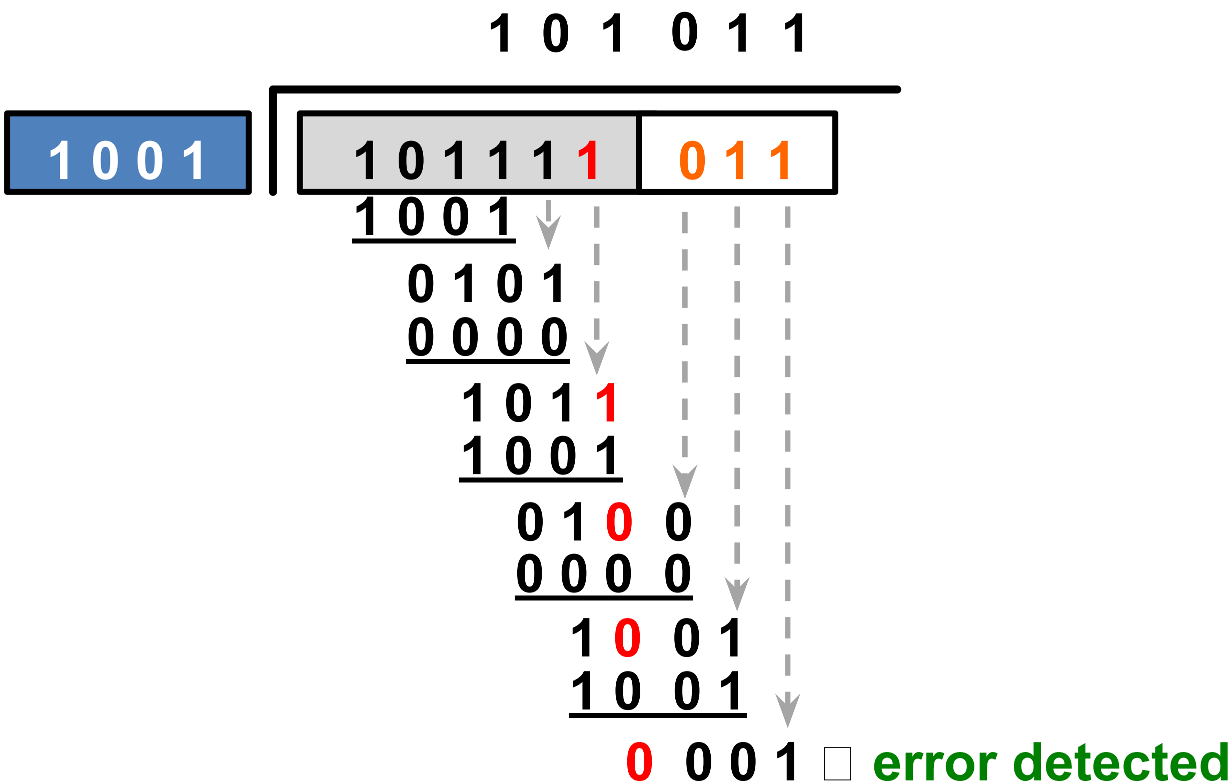
- **Divide received message *C′*(*x*)** by generator *G*(*x*)
  - **If no errors occur,** remainder will be zero



0 0 0 □ no error detected

Detecting errors at the receiver

- **Divide received message *C'(x)*** by generator *G*(*x*)
  - **If errors occur,** remainder **may** be non-zero



Detecting errors at the receiver

- **Divide received message *C′(x)*** by generator *G*(*x*)
  - **If errors occur,** remainder **may** be non-zero

```
              1 0 1  0 1 1
        ___________________
1 0 0 1 | 1 0 1 1 1 1 0 1 0
          1 0 0 1
```

How many errors can the CRC detect?

How do we choose generator G(x)?

```
             0 1 0  0
             0 0 0  0
             --------
               1 0  0 1
               1 0  0 1
               --------
                 0  0 0 0 □ undetected error!
```

Detecting errors with the CRC

- *The **error polynomial*** $E(x) = C(x) + C'(x)$ is the difference between the transmitted and received codeword
  - $E(x)$ tells us which bits the channel flipped

- We can write the **received message $C'(x)$** in terms of $C(x)$ and $E(x)$: $C'(x) = C(x) + E(x)$, so:
  - **Remainder** [$C'(x)$ / $\boldsymbol{G(x)}$] = **Remainder** [$E(x)$ / $\boldsymbol{G(x)}$]

- When does an error go **undetected?**
  - When **Remainder** [$E(x)$ / $\boldsymbol{G(x)}$] = 0

Detecting single-bit errors w/CRC

- Suppose a single-bit error in bit-position i: $E(x) = x^i$
  - Choose $G(x)$ with ≥ 2 non-zero terms: x^{g-1} and 1
  - **Remainder** $[\ x^i / (x^{g-1} + \cdots + 1)\ ] \neq 0$, ***e.g.*:**

$$
\begin{array}{r|l}
 & 1 \\
\hline
1\,0\,0\,1 & 0\,0\,1\,0\,0\,0 \\
 & \underline{1\,0\,0\,1} \\
 & 1
\end{array}
$$

- Therefore a **CRC with this choice of $G(x)$ always detects single-bit errors** in the received message

Error detecting code: CRC

- Far less overhead than error correcting codes
  - Typically **16 to 32 bits** on a **1,500 byte (12 Kbit) frame**

- **Error detecting** properties are **more complicated**

  - But in practice, **“missed” bit errors are exceedingly rare**

Breakout rooms

- What types of errors can be detected?
- Is there a pattern in how errors are detected?

- Extra time: Can you provide an example of an error detection?

Error detecting properties of the CRC

- The CRC will detect:
  - ✔ All **single-bit errors**
    - Provided *G(x)* has two non-zero terms

Error detecting properties of the CRC

- The CRC will detect:
  - ✔ All **single-bit errors**
    - Provided $G(x)$ has two non-zero terms
  - – All **burst errors** of **length ≤ $g - 1$**
    - Provided $G(x)$ begins with x^{g-1} and ends with 1
    - Similar argument to previous property
  - – All **double-bit errors**
    - With conditions on the frame length and choice of $G(x)$
  - – Any **odd number of errors**
    - Provided $G(x)$ contains an even number of non-zero coefficients
- Pattern: errors that manifest as remainders are detected