

# Дәріс 10. Циклдік кодтардың сипаттамасы. Кодтау және декодтау алгоритмдерін іске асыру

Дәріскер: PhD Темирбекова Ж.Е.

# Циклдік кодтарға кіріспе

- ▶ Циклдік кодтар — қателерді анықтау және түзетуге арналған кодтар.
  - ▶ Циклдік ығысу қасиеті: кез келген кодтық комбинацияның барлық ығысқан түрі де кодқа жатады.
  - ▶ Қарапайым, сенімді және тиімді.

# Циклдік кодтардың қолданылуы

- ▶ Деректерді қорғау және қателерді анықтау.
  - ▶ Желілер, байланыс жүйелері, деректерді сақтау құрылғылары.
  - ▶ Аппараттық және бағдарламалық түрде оңай іске асырылады.

# Биттік тізбек және көпмүше ретінде көрсету

- ▶ Циклдік кодтар биттерді екілік көпмүше түрінде қарастырады.
  - ▶ Бұл тәсіл есептеулерді жеңілдетеді.
  - ▶ Әр код сөз — көпмүше: мысалы,  $1011 \rightarrow x^3 + x + 1$ .

# Генератор көпмүшесі деген не?

- ▶ Генератор көпмүшесі кодты анықтайды.
  - ▶ Барлық кодтық сөздер  $g(x)$  көпмүшесіне бөлінеді.
  - ▶  $g(x)$  — циклдік кодтың орталық элементі.

# Циклдік кодтың негізгі қасиеттері

- ▶ 1) Сызықтық қасиетке ие.
  - ▶ 2) Циклдік ығысу кезінде кодтық құрылым сақталады.
  - ▶ 3) Кодтау мен декодтау операциялары жылдам.

# Кодтау процесі түсінігі

- ▶ Хабарлама соңына бақылау биттері қосылады.
  - ▶ Бақылау биттері генератор көпмүшесімен бөліну арқылы табылады.
  - ▶ Нәтижесі — кодталған хабарлама.

# Кодтау алгоритмі (қарапайым түрі)

- ▶ 1. Хабарламаға  $(n-k)$  нөл қосу.
- ▶ 2. Нәтижені  $g(x)$ -ке бөлу.
- ▶ 3. Қалдық — бақылау биттері.
- ▶ 4. Хабарламаға қосып, кодтық сөз алу.

# Кодтау мысалы

- ▶ Мысалы:  $m=1101$ ,  $g=1011$ .
  - ▶  $m \rightarrow 1101000$  (үш нөл қосылды).
  - ▶  $1101000 \div 1011 \rightarrow$  қалдық.
  - ▶ Қалдықты қосып кодтық сөз алынады.

# Декодтау процесі

- ▶ Қабылданған код қайтадан  $g(x)$ -ке бөлінеді.
  - ▶ Қалдық  $= 0 \rightarrow$  қате жоқ.
  - ▶ Қалдық  $\neq 0 \rightarrow$  қате анықталды.

# DivideModulo функциясы

- ▶ Екілік XOR негізінде орындалады.
  - ▶ Бөлу — көпмүшелерді екілік түрде шамалап орындау.
  - ▶ Машинада өте тез орындалады.

# CRC кодтары

- ▶ CRC — циклдік кодтардың арнайы түрі.
  - ▶ Тек қателерді анықтайды, түзетпейді.
  - ▶ Желілер, Ethernet, файлдық жүйелерде қолданылады.

# CRC мысалы

- ▶ message = 11010011101100
  - ▶ generator = 1011
  - ▶ Кодтау:  $\text{message} \cdot (x^{\deg g}) \div g$
  - ▶ Декодтау:  $r \div g \rightarrow \text{syndrome}$ .

# Циклдік кодтардың артықшылықтары

- ▶ Аппараттық іске асыруға өте қолайлы.
  - ▶ Қарапайым XOR операцияларына негізделеді.
  - ▶ Жоғары сенімділік береді.

# Циклдік кодтардың кемшіліктері

- ▶ Барлық түрлері қателерді түзете алмайды.
  - ▶ Кейбір күрделі кодтарда генераторды таңдау қиын.
  - ▶ Көпразрядты қателерді анықтауда шектеулер бар.

# Қорытынды

- ▶ Циклдік кодтар — заманауи байланыс жүйелерінің негізі.
  - ▶ CRC — интернет пен желілердегі қателерді бақылаудың стандарты.
  - ▶ Теориясы қарапайым, қолданылуы кең.
  - ▶ Инженерлер үшін маңызды міндет — генераторды дұрыс таңдау.