ТАҢБАЛЫ ЖӘНЕ ТАҢБАСЫЗ БҮТІН САНДАР

Қосымша код • екілік арифметика • толып кету

UNSIGNED → SIGNED → TWO'S COMPLEMENT → OVERFLOW

Теория: Цифрлық жүйеде бүтін сандар биттер тізбегі түрінде сақталады. Разрядтылық қолжетімді комбинациялар санын анықтайды.

Дәрістің негізгі идеясы

Компьютер «плюс» пен «минусты» емес, биттер тізбегін сақтайды.

11111111

Бұл не: 255 пе, әлде –1 ме?

Жауап осы 8 биттің қалай интерпретацияланатынына байланысты.

БИТТЕР + ДЕРЕК ТҮРІ → МӘН

Интерпретация: Биттердің өздері мәнді анықтамайды. Дерек түрі бағдарламаның бір биттік комбинацияны қалай түсіну керектігін анықтайды.

Unsigned — таңбасыз сандар

Барлық биттер санның мәнін көрсету үшін қолданылады.

n бит үшін: 0 ... 2^n – 1

00000000	00000101	11111111
0	**5**	**255**

8 бит → 0 ... 255

Теория: Unsigned теріс мәндерді қамтымайды. n бит үшін 2^n комбинация бар, сондықтан диапазон 0…2^n–1 болады.

Signed — таңбалы сандар

Оң және теріс мәндерді де көрсету қажет.

n бит үшін: $-2^{n-1} \dots 2^{n-1} - 1$

UNSIGNED

0 ... 255

SIGNED

–128 ... +127

Екі жағдайда да 8 бит 256 комбинация береді.

Теория: Signed оң және теріс мәндерді көрсетуге арналған. Қосымша кодта диапазон: $-2^{n-1} \dots 2^{n-1}-1$.

Бірдей биттер — әртүрлі мәндер



Алдымен дерек түрін білеміз → содан кейін мәнін анықтаймыз

Маңызды: Дерек түрі биттерді өзгертпейді — олардың мағынасын өзгертеді. Сондықтан 11111111_2 255 немесе –1 мәнін білдіруі мүмкін.

Компьютер теріс санды қалай сақтайды?

Таңбалы бүтін сандар үшін әдетте “қосымша код” қолданылады.

+5	00000101	two’s complement	**–5**	11111011

–X алу үшін: биттерді инверсиялау → 1 қосу

Two’s complement: Теріс бүтін сандарды көрсетудің негізгі тәсілі. Оң сандар әдеттегі түрде жазылады; –X алу үшін инверсия жасап, 1 қосады.

Мысал: –5 санын аламыз

1	**+5**	00000101
2	**Инверсия**	11111010
3	**+ 1**	11111011

–5 = 11111011

00000101 → 11111010 → 11111011

Алгоритм: +X санын жазу → 0↔1 инверсиясын орындау → 1 қосу. Мысалы, –3: 00000011 → 11111100 → 11111101.

Қосымша код не үшін қажет?

$$A - B = A + (-B)$$

Азайтуды теріс санды қосу арқылы орындауға болады.

7 – 5	=	7 + (–5)	сумматор

Процессорға бір қосу сұлбасын қолдану оңайырақ

Процессор: A–B=A+(–B). Қосымша кодтың арқасында қосу мен азайтуды процессор ішінде ыңғайлы жүзеге асыруға болады.

Екілік қосу

Ережелер

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 10

тасымал ↑

Мысал: 5 + 3

```
  0101   (5)
+ 0011   (3)
------------
  1000   (8)
```

5 + 3 = 8

Тасымал: $1+1=10_2$: 0 ағымдағы разрядта қалады, ал 1 келесі разрядқа тасымалданады. Бұл ондық жүйедегі тасымалдың екілік аналогы.

Екілік азайту: 7 – 5

7 – 5 = 7 + (–5)

+7 = 00000111
–5 = 11111011

```
  00000111
+ 11111011
----------
1 00000010
```

Нәтиже: 00000010 = 2

Разрядтылық: Амалдар бекітілген разрядтық торда орындалады. Оның шегінен шыққан бит берілген разрядтылықтағы нәтижеге кірмейді.

Толып кету деген не?

Разрядтылық көрсетілетін сандар диапазонын шектейді.

8 биттік unsigned: 0 ... 255

255 + 1 = 256

256 саны 8 битке сыймайды

Бұл — диапазонның толып кетуі.

Overflow: Математикалық нәтиже таңдалған дерек түрінің диапазонына сыймаған кезде толып кету пайда болады.

Unsigned толып кетуі


```
  11111111   (255)
+ 00000001     (1)
-----------------
1 00000000
```

8 битте қалады

00000000

= 0

255 + 1 → 0

Unsigned overflow: uint8 үшін 255-тен кейінгі келесі нәтиже 8 бит деңгейінде 0 болады. Шығыс тасымалы Carry жалаушасымен белгіленуі мүмкін.

Signed толып кетуі

8 биттік signed: –128 ... +127

```
  01111111   (+127)
+ 00000001     (+1)
----------------------
  10000000
```

10000000

signed ретінде = –128

127 + 1 → –128 → OVERFLOW

Signed overflow: int8 үшін 127+1 диапазоннан шығады. 10000000 коды –128 ретінде интерпретацияланады — бұл таңбалы толып кету.

Қорытынды сұлба



Байланыс: Түр → интерпретация тәсілі → қосымша код → екілік арифметика → рұқсат етілген диапазонды бақылау.

Нені есте сақтау керек

1. Unsigned: 8 бит → 0...255
2. Signed: 8 бит → –128...+127
3. **–X: биттерді инверсиялау + 1**
4. **Азайту: A – B = A + (–B)**
5. **Нәтиже диапазоннан тыс → Overflow**

Компьютер биттерді сақтайды. Дерек түрі олардың мәнін анықтайды.

Қосымша теориялық ұғымдар

Разрядтылық — санды сақтау үшін бөлінген биттер саны. Разрядтылық неғұрлым жоғары болса, қолжетімді диапазон соғұрлым кең болады.

MSB және LSB — MSB ең үлкен, сол жақтағы бит; LSB ең кіші, оң жақтағы бит.

Сан коды — цифрлық құрылғы ішінде санды көрсететін биттік комбинация.

Интерпретация — биттік комбинацияны нақты сандық мәнге түрлендіру ережесі.

Диапазондар неге дәл осындай?

Unsigned — 2^n комбинация 0-ден $2^n–1$-ге дейін бөлінеді.

Signed — қосымша кодта диапазон $–2^{n-1}…2^{n-1}–1$ болады.

8 бит — unsigned: 0…255; signed: –128…+127.

Ерекшелік — two’s complement жүйесінде бір ғана нөл бар, сондықтан теріс мәндер саны біреуге артық.

Теріс кодты қалай оқимыз?

1-қадам — егер ең үлкен бит 1-ге тең болса, код signed two’s complement жүйесінің теріс аймағында орналасқан.

2-қадам — барлық биттерді инверсиялау: 0↔1.

3-қадам — 1 қосып, алынған шаманы анықтау.

Мысал — 11110110 → 00001001 → +1 → 00001010 = 10, демек бастапқы сан –10-ға тең.

Carry және Overflow — бір ұғым емес

Carry — ең үлкен разряд шегінен шығатын тасымал; unsigned арифметикасында әсіресе маңызды.

Overflow — signed нәтижесі дерек түрінің рұқсат етілген диапазонынан шықты.

Carry мысалы — 255+1: 11111111+1 → 1 00000000.

Overflow мысалы — int8 жүйесінде 127+1 нәтижесі 10000000 болады, ол –128 ретінде интерпретацияланады.

Микроконтроллерлермен байланысы

uint8_t — 8 биттік unsigned: 0…255. Мысалы, толу деңгейі 0…100 %.

int8_t — 8 биттік signed: –128…127. Мысалы, 0 °C-тан төмен және жоғары температура.

ALU — процессордың арифметикалық-логикалық құрылғысы екілік арифметиканы орындайды.

Практикалық мағынасы — дұрыс дерек түрі өлшенетін шаманың диапазоны мен интерпретациясын дұрыс көрсету үшін қажет.