

2 С# тіліндегі биттік операциялар

2.1 Санақ жүйелері

Ақпаратты көрсетудің негізгі екі тәсілі бар – үзіліссіз немесе аналогтық формада және дискреттік немесе цифрлік формалардың көмегімен ақпаратты көрсету.

Үзіліссіз форма тірі табиғатта ақпаратты бейнелеудің негізгі формасы болы табылады, мысалы, дыбыстық сигналдар, дәмдік сапалар, ауру сезімдері, түс реңдер және т.б., міне бұлардың барлығы да аналогтық сигналдар болып табылады. Көбіне, үзіліссіз сигналдар жазықтықта уақытқа тәуелді функцияның координаттарында үзіліссіз графиктер ретінде бейнеленеді.

Ақпаратты бейнелеудің дискреттік формасы көбінесе техникалық құрылғыларда қолданылады, мысалы, автоматика жүйелерінде, компьютерлерде. Уақытқа тәуелді функция графигінде дискреттік форма деңгейлердің «қабаттары» сияқты түрде бейнеленеді, әрі бір деңгейден келесі деңгейге өту өте жылдам уақыт (лезде) ішінде орындалады, мысалы, қандай да бір орындалушы механизмінің жұмысы екі деңгей – қосулы немесе өшулі деңгейлерімен көрсетіле алады.

Сигналдың аналогтық формасының ақпараттылығы аса жоғары (үзіліссіз функциялар өлшеніп отырған дәлдікке байланысты миллиардтаған түрлі мәндері қабылдай алады), бірақ кедергілерден қорғалу (помехозащищенностью) деңгейі өте төмен болады (сыртқы кедергілердің әсер етуі сигнал формасының аздап өзгерілуіне әкелуі мүмкін). Бұл әсерлердің тірі табиғат үшін еш мәні жоқ, бірақ техникалық құралдар үшін бұл сияқты кедергілер болмауы керек.

Сондықтан, техникалық құрылғыларда, мысалы компьютерлерде ақпаратты бейнелеудің тек дискреттік түрі қолданылады, олардың ішінде ең көп қолданысқа ие болғаны екілік форма (кедергілерден қорғалу деңгейі жоғары болғандықтан).

Екілік формада ақпарат екі деңгейдің көмегімен беріледі, олар – нөл және бір (сигнал бар, сигнал жоқ). Сигналды бейнелеудің екілік формасының кедергілерден қорғалу деңгейі ең жоғары деп саналынады.

Компьютерлерде ақпарат бірлігі ретінде бит (бір разряд) қолданылады, онда 0 немесе 1 жазуға болады. Сегіз бит бір байтты құрайды. Байт – көптеген компьютерлердегі адрестелінетін минималды ақпарат бірлігі. Ақпараттың тағы бір өлшемі ол сөз. 32 разрядтық компьютерлер үшін сөзде 32 бит немесе 4 байт бар, ал 64 разрядтық компьютерлер үшін сөзде 64 бит немесе 8 байт бар,

Компьютердегі басқа да операциялардың ішінде ақпаратты өңдеу кейбір есептеулерді орындауды да қарастырады. Сондықтанда, компьютерге «түсінікті» санақ жүйесін қолдану қажеттілі туындайды. Бұл сияқты мақсаттар үшін сандар тек 0 және 1 мәндерін қабылдай алатын екілік санақ жүйесі сай болды.

Жалпы алғанда, компьютер жадысында сандарды өңдеген уақытта бейнелеудің позициялық (орындық, позиционная) формасы қолданылады – сан (0 немесе 1) сандарының тізбегі ретінде беріледі, олардың әрқайсысының өз орынының санақ жүйесінің негізіне бөлінетін үлесі бар. (При обработке чисел в памяти компьютера обычно используется позиционная форма представления – число представляется в виде последовательности цифр (0 или 1), позиция каждой из которых имеет свой вес кратный основанию системы счисления.) Мысалы, 1001 екілік санына бейнелеудің позициялық формасындағы $1*2^3 + 0*2^2 + 0*2^1 + 1*2^0$ жазбасы сәйкес келеді. Егер есептеулерді орындасақ 9 санын аламыз.

Сандардың позициялық формасы нақты сандарды жазуда да қолданылады, олардың бөлшек жақтарының ретінің мәні теріс болады. Мысалы, 45,36 ондық саны позициялық формада $4*10^1 + 5*10^0 + 3*10^{-1} + 6*10^{-2}$ түрінде жазылады. Сәйкесіше, нақты сандар екілік санақ жүйесінде де бейнеленіледі.

Сонымен, екілік санақ жүйесі сандарды компьютерге «ұғынықты» формада бейнелеуге мүмкіндік береді.

Екілік сандарды экранда бейнелеу көп орынды талап етеді, сондықтан екіге бөлінетін санақ жүйелерін – сегіздік немесе он алтылық санақ жүйелерін қолдану керек деп шешілген.

Сандарды сегіздік жүйеде жазу үшін 0–ден 7–ге дейінгі 8 сан қажет. Үш екілік санның тізбегінің сегіз түрлі комбинациясы бар болғандықтан, екілік сандардың әр үйлесімін (триадасын) сегіздік санның біреуінің көмегімен бейнелеуге болады:

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

Мысалы, 01011110 екілік санын ыңғайлы болу үшін 01 011 110 триадаларының көмегімен жаамыз да, сегіздік санақ жүйесінде бейнелейміз 136, ал ол болса ондық жүйеде $1*8^2 + 3*8^1 + 6*8^0 = 94$ тең.

Санды он алтылық жүйеде жазу үшін 0–ден 9–ге дейінгі сандар және латын әліпбиінің А, В, С, D, E, F әріптері 16 сан қажет. Төрт екілік санның тізбегінің 16 түрлі комбинациясы бар болғандықтан, екілік сандардың әр үйлесімін (тетрадасын) ол алтылық санның біреуінің көмегімен бейнелеуге болады:

0000 – 0	1000 – 8
0001 – 1	1001 – 9
0010 – 2	1010 – А
0011 – 3	1011 – В
0100 – 4	1100 – С

0101 – 5 1101 – D
0110 – 6 1110 – E
0111 – 7 1111 – F

Мысалы, 01011110 санын ыңғайлы болу үшін 0101 1110 татрадаларының көмегімен жазамыз және оны алтылық жүйеде бейнелейміз 5E, ал ол болса ондық санақ жүйесінде $5 \cdot 16^1 + 14 \cdot 16^0 = 94$ тең.

Бағдарламалаушылар компьютерлік техникамен жұмыс жасағанда көбінесе, он алтылық санақ жүйесін қолданады, өйткені 1 байт ақпаратты екі он алтылық санмен бейнелеген оңай.

C# тілінде санды екілік немесе сегіздік санақ жүйесінде көрсету формалары жоқ, бірақ он алтылық санақ жүйесі кеңінен берілген.

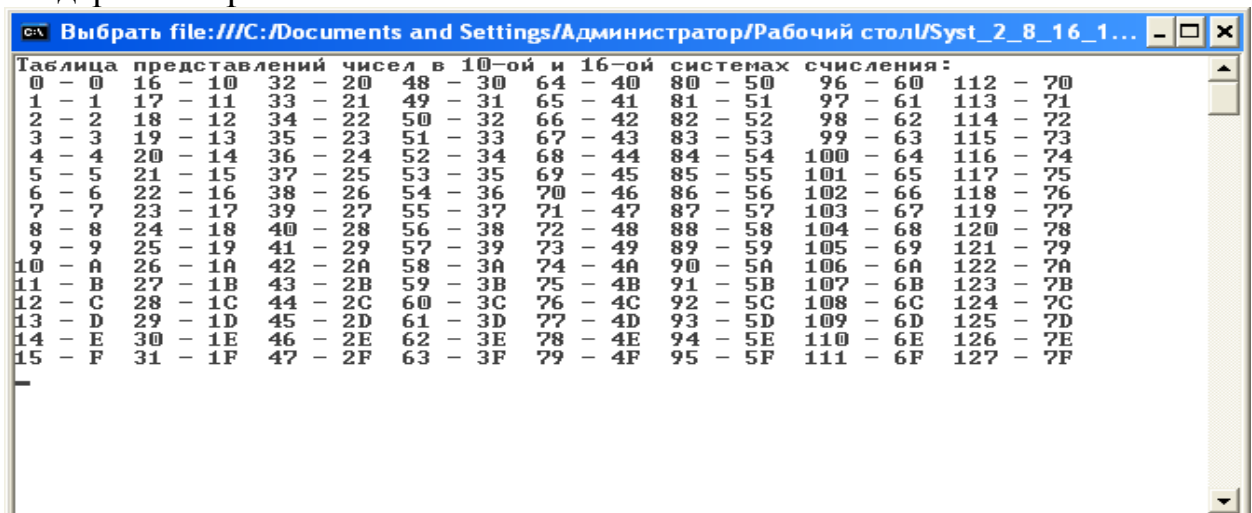
Ондық санақ жүйесінің 0-ден 217-ге дейінгі сандарын он алтылық санақ жүйесіне түрлендірудің бағдарламалық жүзеге асырылуы берілген.

Бағдарламаның бастапқы коды:

```
using System;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            uint i;
            Console.WriteLine("Таблица представлений чисел в 10-ой и 16-ой системах счисления:");
            for (i = 0; i <= 15; i++)
                Console.WriteLine("{0,2} - {1:X} {2} - {3:X} {4} - {5:X} {6} - {7:X} {8} - {9:X} {10} - {11:X} {12,3} - {13:X} {14,3} - {15:X}", i, i, i+16, i+16, i+32, i+32, i+48, i+48, i+64, i+64, i+80, i+80, i+96, i+96, i+112, i+112);
            Console.ReadLine();
        }
    }
}
```

Бағдарлама жұмысы:



2.1-сурет – Ондық сандарды он алтылық сандарға түрлендіру

Егер екілік сандардың бейнесін «таза» түрде алу керек болса, онда әдетте ол үшін сандарды бейнелеудің «бағдарламалық» түрде толтырылатын жолдық формасы қолданылады, мысалы:

```
using System;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            uint i;
            string d;
            char[] b = new char[8] {'0','0','0','0','0','0','0','0'};
            Console.WriteLine("Таблица представлений чисел в 10-ой, 2-ой
и 16-ой системах счисления:");
            for (i = 0; i <= 63; i++)
            {
                d = "";
                for (int j = 0; j < 8; j++) d = d + b[j];
                Console.WriteLine("{0,2} - {1} - {2:X} ", i, d, i);
                if (b[7] == '0') b[7] = '1';
                else {b[7] = '0'; if (b[6] == '0') b[6] = '1';
                    else {b[6] = '0'; if (b[5] == '0') b[5] = '1';
                        else {b[5] = '0'; if (b[4] == '0') b[4] = '1';
                            else {b[4] = '0'; if (b[3] == '0') b[3] = '1';
                                else {b[3] = '0'; if (b[2] == '0') b[2] = '1';
                                    else {b[2] = '0'; if (b[1] == '0') b[1] = '1';
                                        else {b[1] = '0'; if (b[0] == '0') b[0] = '1';
                                            else { b[0] = '0'; }}}}]]]]}}
                Console.ReadLine();
            }
        }
    }
}
```

Таблица представлений чисел в 10-ой, 2-ой и 16-ой системах счисления:

0 - 00000000 – 0	17 - 00010001 – 11	34 - 00100010 – 22	51 - 00110011 - 33
1 - 00000001 – 1	18 - 00010010 – 12	35 - 00100011 – 23	52 - 00110100 - 34
2 - 00000010 – 2	19 - 00010011 – 13	36 - 00100100 – 24	53 - 00110101 - 35
3 - 00000011 – 3	20 - 00010100 – 14	37 - 00100101 – 25	54 - 00110110 - 36
4 - 00000100 – 4	21 - 00010101 – 15	38 - 00100110 – 26	55 - 00110111 - 37
5 - 00000101 – 5	22 - 00010110 – 16	39 - 00100111 – 27	56 - 00111000 - 38
6 - 00000110 – 6	23 - 00010111 – 17	40 - 00101000 – 28	57 - 00111001 - 39
7 - 00000111 – 7	24 - 00011000 – 18	41 - 00101001 – 29	58 - 00111010 - 3A
8 - 00001000 – 8	25 - 00011001 – 19	42 - 00101010 - 2A	59 - 00111011 - 3B
9 - 00001001 – 9	26 - 00011010 - 1A	43 - 00101011 - 2B	60 - 00111100 - 3C
10 - 00001010 – A	27 - 00011011 - 1B	44 - 00101100 - 2C	61 - 00111101 - 3D
11 - 00001011 – B	28 - 00011100 - 1C	45 - 00101101 - 2D	62 - 00111110 - 3E

12 - 00001100 – C 29 - 00011101 - 1D 46 - 00101110 - 2E 63 - 00111111 - 3F
13 - 00001101 – D 30 - 00011110 - 1E 47 - 00101111 - 2F
14 - 00001110 – E 31 - 00011111 - 1F 48 - 00110000 - 30
15 - 00001111 – F 32 - 00100000 – 20 49 - 00110001 - 31
16 - 00010000 – 10 33 - 00100001 – 21 50 - 00110010 - 32

Бағдарлама шынайы кестенің барлық мәндерін бір ұзын бағанға шығарады, бірақ дәрісте орынды үнемдеу үшін кесте 4 бағанмен берілген.

2.2 C# тілінің разрядтық логикалық операциялары

Разрядтық логикалық операциялары разрядтық логикалық көбейту немесе конъюнкция, разрядтық логикалық қосу немесе дизъюнкция және разрядтық логикалық шығару немесе НЕМЕСЕ-ні шығару операцияларымен берілген. Сонымен қатар, логикалық операцияларға разрядтық логикалық терістеу немесе толықтыру операциясы да жатады.

Екі санмен разрядтық логикалық көбейту операциясын (& белгіленеді) орындағанда олардың биттік (побитовое) «логикалық көбейтуі» орындалады да, нәтиже сәйкес битке жазылады. Әрине, логикалық көбейтудің ережесіне сәйкес егер операцияға қатысатын екі битте 1-ге тең болса, нәтиже 1-ге тең болады.

Екі санмен разрядтық логикалық қосу операциясын (| белгіленеді) орындағанда олардың биттік (побитовое) «логикалық қосылуы» орындалады да, нәтиже сәйкес битке жазылады. Әрине, логикалық қосудың ережесіне сәйкес егер операцияға қатысатын екі биттің тым болмағанда біреуі 1-ге тең болса, нәтиже 1-ге тең болады.

Екі санмен разрядтық логикалық шығару, НЕМЕСЕ-ні шығаратын операциясын (^ белгіленеді) орындағанда олардың биттік салыстырылуы орындалады да, және операцияға қатысатын екі биттің тек біреуі ғана 1-ге тең болса, нәтиже 1-ге тең болады.

Разрядтық логикалық терістеу, немесе толықтыру операциясын (~ белгіленеді) орындағанда операцияға қатысатын санның барлық разрядында бірлер нөлдермен, ал нөлдер бірлермен алмастырылады.

Материалды бекітіп алу үшін оқу бағдарламасының жұмысын қарастырайық, мұнда аталған барлық разрядтық логикалық операцияларды қолданамыз.

```
using System;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main()
        {
```

```

uint a, b, c;
Console.Write("Введите целое значение a ");
a = Convert.ToUInt32(Console.ReadLine());
Console.Write("Введите целое значение b ");
b = Convert.ToUInt32(Console.ReadLine());
c = a & b;
Console.WriteLine("a & b = {0} & {1} = {2}", a, b, c);
Console.WriteLine("ax & bx = {0:X} & {1:X} = {2:X}", a, b, c);
c = a | b;
Console.WriteLine("a | b = {0} | {1} = {2}", a, b, c);
Console.WriteLine("ax | bx = {0:X} | {1:X} = {2:X}", a, b, c);
c = a ^ b;
Console.WriteLine("a ^ b = {0} ^ {1} = {2}", a, b, c);
Console.WriteLine("ax ^ bx = {0:X} ^ {1:X} = {2:X}", a, b, c);
Console.WriteLine("~a = {0} ~b = {1}", ~a, ~b);
Console.WriteLine("~ax = {0:X} ~bx = {1:X}", ~a, ~b);
Console.ReadKey();
}
}
}

```

Бағдарлама жұмысы:

Введите целое значение a 42

Введите целое значение b 23

$a \& b = 42 \& 23 = 2$

$ax \& bx = 2A \& 17 = 2$

$a | b = 42 | 23 = 63$

$ax | bx = 2A | 17 = 3F$

$a \wedge b = 42 \wedge 23 = 61$

$ax \wedge bx = 2A \wedge 17 = 3D$

$\sim a = 4294967253 \quad \sim b = 4294967272$

$\sim ax = FFFFFFFD5 \quad \sim bx = FFFFFFFE8$

Разрядтық логикалық көбейту операциясының жұмысын тексерейік, а және b сандарының мәндерін екілік санақ жүйесінде бейнелейік:

$ax = 2A = 0010\ 1010$ – мәндер дәрістің алдыңғы пунктiнiң бағдарлама

$bx = 17 = 0001\ 0111$ жұмысының нәтижесiнен алынған

$ax \& bx = 0000\ 0010$ – нәтиже ондық (және он алтылық) 2 санына тең.

2.3 C# тiлiндегi биттiк жылжыту операциялары

Сандарды биттiк жылжыту операциялары түрлi алгоритмдерде қолданылады, мысалы биттер тiзбегiн параллельдi кодқа түрлендiруде және керiсiнше параллельдi кодты биттер тiзбегiне түрлендiруде қолданылады. Компьютердiң кейбiр құрылғылары бұл алгоритмдердi арнайы жылжыту регистрлерiнiң көмегiмен аппаратты түрдi жүзеге асырады, мысалы, магниттiк немесе оптикалық дискiлерден ақпаратты оқу және жазу сұлбасы

(схема). Кейбір жағдайларда бұл алгоритм бағдарламалық түрде жүзеге асырылады, мысалы, USB арқылы қандай да бір мәліметтерді жіберу хаттамасын (протокол передачи данных) бағдарламалық түрде жүзеге асыру кезінде.

Екілік санды 2-ге көбейту арифметикалық операциясы екілік санды сол жаққа қарай бір разрядқа жылжытқанға **пара пар (эквивалента)**, ал 2-ге бөлу операциясы – екілік санды бір разрядқа оңға жылжытқанға **пара пар** болады.

C# тілінде жылжыту операцияларын бағдарламалық түрде жүзеге асыру үшін екі операция бар:

- Екілік санды солға жылжыту операциясы (<< белгіленеді);
- Екілік санды оңға жылжыту операциясы (>> белгіленеді).

Екі операцияның да жазылу формалары бір-біріне ұқсас, операция белгісіне дейін жылжыту жасалынатын сан көрсетіледі, ал операциядан соң жылжыту жасалынатын разрядтар санын көрсеті керек, мысалы:

```
c = 5 << 8;           // a санын солға екі разрядқа жылжыту.
```

```
c = b >> 1;          // b санын оңға бір разрядқа жылжыту.
```

Жылжыту операциялары тек бүтін типті сандармен жұмыс істеуге арналған, мысалы, int, uint және т.б.

Солға жылжытқанда босаған разрядтар нөлге айналады (**затвигается 0**).

Оңға жылжытқанда босаған разрядтар нөлге айналады, бірақ жылжыту санның таңбасын сақтай отыра жүргізіледі.

Мысал ретінде он алтылық 5 санын (екілік 0000 0000 0101 саны) 8 разрядқа солға жылжытуды қарастырамыз – нәтижесінде екілік 0101 0000 0000 саны немесе он алтылық 500 саны шығу керек.

Бағдарлама коды:

```
using System;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main()
        {
            uint a;
            Console.WriteLine("Введите целое значение a ");
            a = Convert.ToUInt32(Console.ReadLine());
            for (int i = 1; i <= 8; i++)
            {
                a = a << 1;
                Console.WriteLine("ax = {0:X}      {1}", a, a);
            }
            for (int i = 4; i > 0; i--)
            {
                a = a >> 2;
                Console.WriteLine("ax = {0:X}      {1}", a, a);
            }
            Console.ReadKey();
        }
    }
}
```

}

}

Бағдарлама жұмысы:

Введите целое значение a 5

ax = A 10

ax = 14 20

ax = 28 40

ax = 50 80

ax = A0 160

ax = 140 320

ax = 280 640

ax = 500 1280

ax = 140 320

ax = 50 80

ax = 14 20

ax = 5 5

a санын солға жылжыту циклда жылжыту қадамы бірге тең етіп жүргізіледі. Циклдің жұмыс нәтижесі санның әр қадамнан соң 5-тен 1280-ге дейін екі еседен өсіп отырғанын көрсетеді.

Санды оңға жылжыту да циклда жүргізілді, ал жылжыту қадамы 2 деп берілді. Бұл циклдің жұмыс нәтижесі a санының әр қадамнан соң 4 есеге азайып отырғанын көрсетеді.