1 Bináris számrendszer

1.1 Alapfogalmak

. Számjegy: Számok reprezentálására használt szimbólum.

. Számrendszer (számábrázolási rendszer): Megadja a számok egy csoportjának ábrázolási szabályait, vagyis, hogy adott számjegyek sorozata milyen számot jelenít meg. Minden számhoz egyedi ábrázolást rendel.

. Helyiértékes számrendszer: A számjegy maga csak *alaki érték*et jelöl, a valódi érték a számjegy relatív pozíciójától függ. Ilyen a tízes és a kettes számrendszer is.

Ellenpélda: római számrendszer, unáris számrendszer [1]

. Alapszám: Egy b alapszámú helyiértékes számrendszer az első b darab természetes szám ábrázolására használ különböző szimbólumokat. A további számokat a helyiértékek segítségével jelöli. A helyiértékek az alapszám hatványai.

1.2 Példák

Megjegyzés: bármely r valós szám esetén $r^0 = 1$. Erre a következő példák megértéséhez van szükség.

1. Példa
A tízes számrendszer 10 különböző számjegyet használ 0-9-ig. A következő, tízes számrendszer-beli szimbólumokat így értelmezzük:

$10 = 0 \times 10^0 + 1 \times 10^1$
$100 = 0 \times 10^0 + 0 \times 10^1 + 1 \times 10^2$

Most nézzünk egy bonyolultabb számot:
$2536 = 6 \times 10^0 + 3 \times 10^1 + 5 \times 10^2 + 2 \times 10^3$

A helyiértékek (egy, tíz, száz, ezer, stb.) a 10 hatványai, ezeket szorozzuk be a helyiértéken álló számjeggyel. A helyiértékek jobbról balra olvasva nőnek.

2. Példa
A kettes számrendszer csak két számjegyet használ, a 0-át és az 1-et. A szimbólumokat a fentiekkel analóg módon értelmezzük, de itt a helyiértékek a 2 hatványai:

$1010 = 0 \times 2^0 + 1 \times 2^1 + 0 \times 2^2 + 1 \times 2^3$, ami $2 + 8 = 10$ -nek felel meg a tízes számrendszerben.

[1] Az unáris számrendszer egy szimbólum ismétlésével jelöli a számokat, pl. $5 = |||||$

Ugyanez egy bonyolultabb számmal:
$100101 = 1 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 0 \times 2^3 + 0 \times 2^4 + 1 \times 2^5 = 1 + 4 + 32 = 37$

Általánosan: b alapszámú számrendszerben, ha a_k jelöli a k-adik helyiértéken lévő számjegyet, a számjegyeket így értelmezzük:

$(a_n a_{n-1} ... a_1 a_0)_b = \sum_{k=0}^{n} a_k b^k$ (a k-adik helyiértéken álló számjegyet az alapszám k-adik hatványával szorozzuk be, és a szorzatokat összeadjuk).

1.3 Átalakítás kettes számrendszerbe

A számokat tízes számrendszerből kettes számrendszerbe **maradékos osztás** segítségével alakítjuk át. Először nézzünk egy példát a 10-et használva osztónak.

osztandó	osztó	hányados	maradék
2536	10	253	6
253	10	25	3
25	10	2	5
2	10	0	2

A maradékokat visszafelé olvasva kapjuk meg az eredeti számot!

Most a 2-est használjuk osztónak:

osztandó	osztó	hányados	maradék
37	2	18	1
18	2	9	0
9	2	4	1
4	2	2	0
2	2	1	0
1	2	0	1

Az előzőhöz hasonlóan, a maradékokat visszafelé olvasva kapjuk meg a kettes számrendszerbeli reprezentációt.

2 A számítógép működési elve

Az ítéletlogikai példákban állításokhoz rendeltünk logikai változókat. Egy logikai változó azonban jelentheti például azt is, hogy adott vezeték áram alatt van-e (ha igen, akkor a változó IGAZ, különben HAMIS). A számítógép ilyen áramkörökből épül fel, és legelemibb szinten csak annyit tud, hogy az adott vezetéket áram alá helyezi, vagy éppen nem. Ennyi információból kell egy sokkal komplexebb rendszert felépíteni. Mindezt logikai műveletek segítségével teszi.

Példa: Half Adder

A *half adder* (vagy *félösszeadó*) két bináris számot ad össze, mindkettő értéke csak 0 vagy 1 lehet.

- $0 + 0 = 00$
- $0 + 1 = 01$
- $1 + 0 = 01$
- $1 + 1 = 10$

A 0-át és az 1-et értelmezhetjük logikai változóként is: mindkettő egy vezeték, és ha adott vezetéken megy áram, akkor az érték 1, különben 0. Olyan logikai műveletekkel kell ezeket összekapcsolnunk, hogy a fenti eredményt kapjuk.

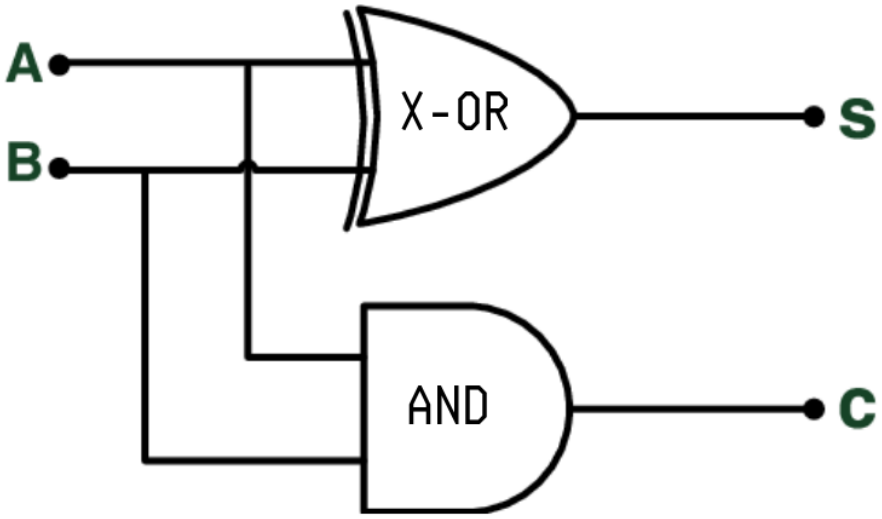
Milyen művelettel kössük össze a két értéket, hogy olyan kifejezést kapjunk, melynek értéke **abban az egy esetben igaz, ha mindkét érték igaz**? Az AND (*és*) művelettel! Ez adja az összeg **első számjegyét**.

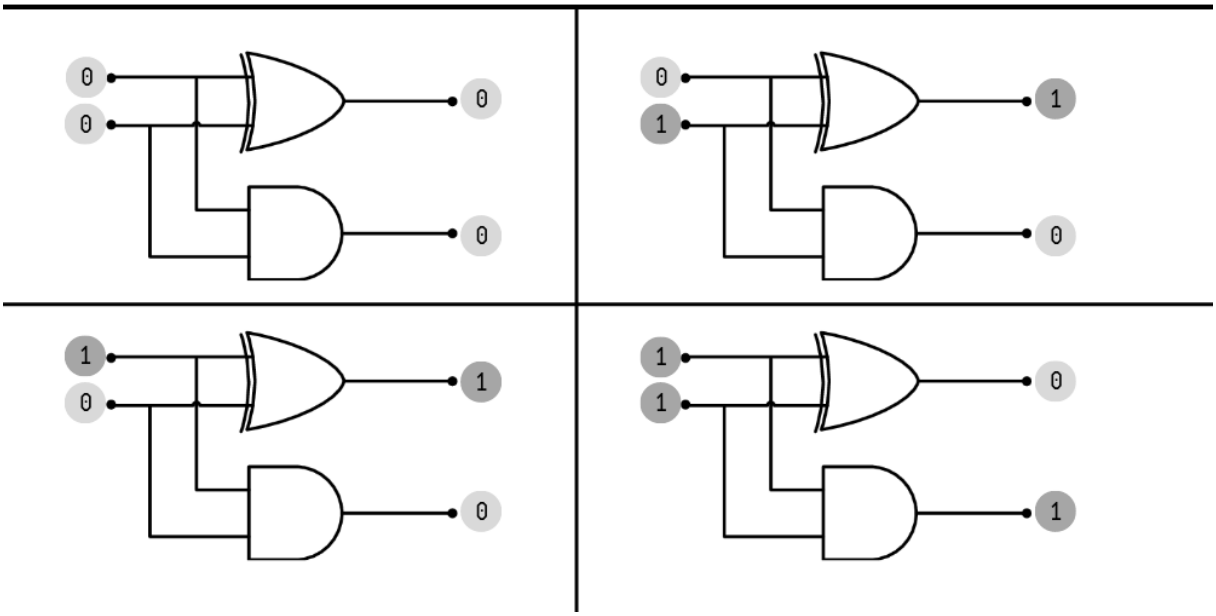
- $0 + 0 = \mathbf{0}0$
- $0 + 1 = \mathbf{0}1$
- $1 + 0 = \mathbf{0}1$
- $1 + 1 = \mathbf{1}0$

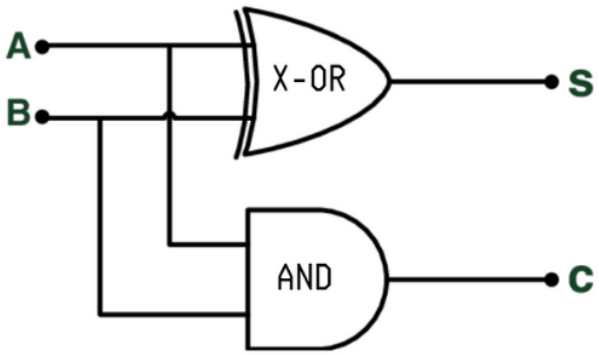
És milyen műveletből áll össze az a kifejezés, amely **abban a két esetben veszi fel az igaz értéket, ha pont az egyik komponense igaz**? Az X-OR (*kizáró vagy*) művelettel! Ez adja az összeg **második számjegyét**.

- $0 + 0 = 0\mathbf{0}$
- $0 + 1 = 0\mathbf{1}$
- $1 + 0 = 0\mathbf{1}$
- $1 + 1 = 1\mathbf{0}$

A lenti ábrák szemléltetik a gép működését.


A
B
X-OR
S
AND
C


0
0
0
0
1
0
1
1
0
1
0
0
1
1
0
1



A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0