Representing numbers in different bases

In base r: $D = a_{n-1}\, a_{n-2} \ldots a_0 \,.\, a_{-1}\, a_{-2} \ldots$

In base 10: $N = a_{n-1} * r^{n-1} + a_{n-2} * r^{n-2} + \ldots + a_0 + a_{-1} * r^{-1} + a_{-2} * r^{-2} + \ldots$

Complement to Base r

Definition:

Number		r-complement
D	xxxxxxxx . yyyyyy (n digits, m digits) →	$r^n - D$
$(1101)_2$, n=4	2-complement →	2^4: 10000-1101 =0011
$(12)_{10}$, n=2	10-complement →	10^2: 100-12 = 88

Complement-1 to Base r

Definition:

Number		(r-1) complement
D	xxxxxxxx . yyyyyy (n digits, m digits) →	$r^n - r^{-m} - D$
$(1101.11)_2$, n=3 m=2	1-complement →	1111.11-1101.11 =0010.00
$(12)_{10}$, n=1	9-complement →	99-12 = 87

Another representation of 2 complement

BCD Weight: 2^{n-1}

2-complement Weight: -2^{n-1}

→ $a_{n-1}\, a_{n-2} \ldots a_0 \,.\, a_{-1}\, a_{-2} \ldots$

BCD Coding

Two complement

1101 = $-2^3 + 2^2 + 1$ → -3

\- 0011 → -3

Calculating the r complement

(r-1) complement: $r^n - r^{-m} - D$ → ($+r^{-m}$) → r complement: $r^n - D$

Number (base 2): 1101

1-complement: 0010 → (+1) → 0011

0 in complement to 1

Number	1-complement
00000 →	11111

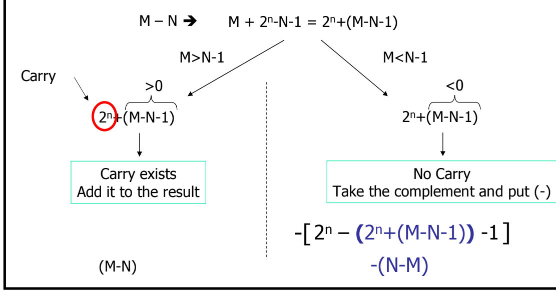
Two representations to 0!

Complement to 1 vs. 2

	1-Complement	2-Complement
Calculation	Easy	Harder
Zero presenrtation	Dual	Singe

We usually use 2-complement

Subtraction using 1-complement



Example I

3	011
-5	+010
No Carry	101
	-010 = -2

Example II

3	011
-2	+101
Carry 1	000
	1
	001

Changing number of bits

Given a number in 2 complement with n bits

What is the representation with m>n bits ?

Changing number of bits

0011 → 00 0011

1011 → 11 0011

Binary Multiplication

```
   1101        13
 X 0011      X 03
 ------
   1101
  1101
------------------
 100111        39
```

2-Complement multiplication

```
X  -3
    5
 ----
```

```
         X  1101
            0101
           ------
          111101
          00000
          1101
Carry    -------
  1       110001
```

2-Complement multiplication

```
X  -3
   -5
 ----
```

```
X  1101
   1011
  ------
  ?????
```

2-Complement multiplication

```
X  -3
   -5
 ----
```

```
                          X  1101
                             1011
                            ------
                          1111101
                          111101
Remember:                 00000
Last digit has
negative weight  ──→      0011
                        ----------
                          0001111
                                =15
```