Digital Logic Design (CSNB163)

Module 2

Recaps…

- In Module 1, we have learned basic number representation which include binary, octal, hexadecimal and decimal.
- In **digital systems**, numbers are manipulated in **binary** format. However, **to ease human-machine interaction,** we describe numbers in **octal** or **hexadecimal.** e.g. $11111111_2 = 377_8 = FF_{16}$
- Where as, **decimal** is used in **daily life**.
- In this module, examples are only in decimal and binary (i.e. to ease learning via decimal examples and relate it to actual digital system implementation via binary examples).

Digital Subtraction Operation

- In digital systems, digital numbers are manipulated via digital logic operation.
- We have learned basic number operation in Module 1 which include subtraction.
- Subtraction operation in digital logic is **NOT** similar to our normal paper and pencil calculation (i.e. **borrow concept is not practical**).
- Instead, it is performed via **complement** function in order to **simplify** digital logic operation, hence to produce **cheaper** circuits.

Complement Function

- There are two types of complements for each radix *r* based system:
  - **Diminished Radix Complement** (*r*-1)' complement
    - E.g. 9's complement for radix 10 (decimal)
    - E.g. 1's complement for radix 2 (binary)
  - **Radix Complement** (*r*)' complement
    - E.g. 10's complement for radix 10 (decimal)
    - E.g. 2's complement for radix 2 (binary)

Diminished Radix Complement

- Formulae:
  - Given a number ***N***
    in base ***r***
    having ***n*** total digits,
    the **(*r*-1)'s complement of *N*** is defined as

$$(r^n - 1) - N$$

Diminished Radix Complement (Example 1)

- Convert 12345_{10} to its 9's complement:
- Working:
  - Given a number ***N* = 12345**
    in base ***r* = 10**
    having ***n* = 5** total digits,
    the ***9*'s complement of 12345** is defined as:

$$
\begin{aligned}
(r^n - 1) - N &= (10^5 - 1) - 12345 \\
&= (100\,000 - 1) - 12345 \\
&= (99\,999) - 12345 \\
&= 87654
\end{aligned}
$$

Diminished Radix Complement (Example 2)

- Convert 1011000_2 to its 1's complement:
- Working:
  - Given a number ***N* = 1011000**

    in base ***r* = 2**

    having ***n* = 7** total digits,

    the ***1*'s complement of 1011000** is defined as:

$$
\begin{aligned}
(r^n - 1) - N &= (2^7 - 1) - 1011000 \\
&= (10000000 - 1) - 1011000 \\
&= (1111111) - 1011000 \\
&= 0100111
\end{aligned}
$$

1's complement is performed by inversing all the bits

Radix Complement

- Formulae:
  - Given a number ***N***
    in base ***r***
    having ***n*** total digits,
    the **(*r*)'s complement of *N*** is defined as

$$[(r^n - 1) - N] + 1 \text{ OR } r^n - N$$

(r-1)'s complement + 1

Radix Complement (Example 1)

- Convert 12345_{10} to its 10's complement:
- Working:
  - Given a number ***N* = 12345**

    in base ***r* = 10**

    having ***n* = 5** total digits,

    the ***10*'s complement of 12345** is defined as:

$$(r^n - 1) - N + 1 = (10^5 - 1) - 12345 + 1$$
$$= (100\,000 - 1) - 12345 + 1$$
$$= (99\,999) - 12345 + 1$$
$$= 87655$$

OR

$$r^n - N = 10^5 - 12345 = 87655$$

Radix Complement (Example 2)

- Convert 1011000_2 to its 2's complement:
- Working:
  - Given a number ***N* = 1011000**
    in base ***r* = 2**
    having ***n* = 7** total digits,
    the ***2*'s complement of 1011000** is defined as:

Short cut 2's complement:
1st inversing all the bits
2nd add 1

$(r^n - 1) - N + 1 = (2^7 - 1) - 1011000 + 1$
$= (10000000 - 1) - 1011000 + 1$
$= (1111111) - 1011000 + 1$
$= 0100111 + 1 = 0101000$

OR

$r^n - N = 2^7 - 1011000 = 10000000 - 1011000 = 0101000$

Subtraction with Complements

- Subtraction $M - N$ can be carried using radix r's complement. In digital system of radix r = 2, subtraction is performed using 2's complement.
- Recaps of radix complement formulae:
  - Given a number ***N*** in base ***r*** having ***n*** total digits, the **(*r*)'s complement of *N*** is defined as

$$r^n - \mathrm{N}$$

- For $M > N$

$$M - N = [M + (r^n - \mathrm{N})] - r^n$$

Subtraction via 10's Complement (Example 1)

- Calculate $56_{10} - 42_{10}$ using 10's complement.
- Working:
  - Given ***M* = 56** and ***N* = 42**
    in base ***r* = 10** having ***n* = 2** total digits

1st:-> get 10's complement of N

r^n = 100
10's complement of N = 100 – 42
= 58

2nd :-> add 10's complement of N to M

M = 56
10's complement of N = 58
M + 10's complement of N = 114

3rd:-> minus r^n from 2nd result

2nd result = 114
r^n = 100
r^n - (2nd result) = 14

Subtraction via 10's Complement (Example 2)

- Calculate $61_{10} - 27_{10}$ using 10's complement.
- Working:
  - Given ***M* = 61** and ***N* = 27**
    in base ***r* = 10** having ***n* = 2** total digits

1st:-> get 10's complement of N

r^n = 100

10's complement of N= 100 – 27

= 73

2nd :-> add 10's complement of N to M

M = 61

10's complement of N = 73

M + 2's complement of N =134

3rd:-> minus r^n from 2nd result

2nd result = 134

r^n = 100

r^n - (2nd result)= 34

Subtraction via 2's Complement (Example 1)

- Calculate $111000_2 - 101010_2$ ($56_{10} - 42_{10}$) using 2's complement.
- Working:
  - Given ***M* = 111000** and ***N* = 101010** in base ***r* = 2** having ***n* = 6** total digits

1st:-> get 2's complement of N

1's complement of N = 010101

2's complement of N = 010110

2nd :-> add 2's complement of N to M

M	=	111000
2's complement of N	=	010110
M + 2's complement of N	=	1001110

3rd:-> minus r^n from 2nd result

2nd result	=	1001110
r^n	=	1000000
r^n - (2nd result)	=	1110

Note: $56_{10} - 42_{10} = 14_{10}$

Subtraction via 2's Complement (Example 2)

- Calculate $111101_2 - 011011_2$ (i.e. $61_{10} - 27_{10}$) using 2's complement.
- Working:
  - Given M = **111101** and N = **011011** in base r = **2** having n = **6** total digits

1st:-> get 2's complement of N
1's complement of N = 100100
2's complement of N = 100101

2nd :-> add 2's complement of N to M

M	=	111101
2's complement of N	=	100101
M + 2's complement of N	=	1100010

3rd:-> minus r^n from 2nd result

2nd result	=	1100010
r^n	=	1000000
r^n - (2nd result)	=	100010

Note: $61_{10} - 27_{10} = 34_{10}$

Digital Logic Design (CSNB163)

End of Module 2