Digital Logic Design

Hajar Falahati

Department of Computer Engineering
IRAN University of Science and Technology

hfalahati@iust.ac.ir

Number System

- Positioned number

$$N = (a_{n-1}a_{n-2} \ldots a_1a_0 \,.\, a_{-1}a_{-2} \ldots a_{-m})_r$$

  - . = radix point
  - r = radix or base
  - n = number of integer digits to the left of the radix point
  - m = number of fractional digits to the right of the radix point
  - a_{n-1} = most significant digit (MSD)
  - a_{-m} = least significant digit (LSD)

- Polynomial notation
  - Series representation

$$N = a_{n-1} \times r^{n-1} + a_{n-2} \times r^{n-2} + \ldots + a_0 \times r^0 + a_{-1} \times r^{-1} \ldots + a_{-m} \times r^{-m}$$

$$\sum_{i=-m}^{n-1} a_i r^i$$

Outline

- Number Systems
  - Information representation
  - Base Conversion
  - Conversion from base r to decimal
  - Conversion from decimal to base r
  - Special Conversion
- Arithmetic Operations
  - Addition
  - Subtraction
  - Multiplication
  - Division
- Signed and Unsigned Numbers

Special Conversion

Binary → Octal

- Binary digits are **grouped** by **threes**
  - Starting from the least significant bit
  - Proceeding from the **left** and to the **right**.
- Fill out the last group of three if necessary
  - Add leading 0s (or trailing zeros)
- Replace each three bits with the equivalent octal digit

Binary to Decimal: Sample 1

- $(11111110)_2 = (\ ?\ )_8$

Digit notation	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0
Digit	1	1	1	1	1	1	1	0
3-bit groups								
Octal digit								

Binary to Decimal: Sample 1

- $(11111110)_2 = (\ ?\ )_8$

Digit notation	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0
Digit	1	1	1	1	1	1	1	0
3-bit groups	11		111			110		
Octal digit								

Binary to Decimal: Sample 1

- $(11111110)_2 = (\ ?\ )_8$

Digit notation	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0
Digit	1	1	1	1	1	1	1	0
3-bit groups	011		111			110		
Octal digit								

Binary to Decimal: Sample 1

- $(11111110)_2 = (\ 376\ )_8$

Digit notation	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0
Digit	1	1	1	1	1	1	1	0
3-bit groups	011		111			110		
Octal digit	3		7			6		

Octal → Binary

- Replace each octal digit by three bits, binary
- $(376.22)_8 = (011111110.010010)_2$

Octal digit notation	O_2			O_1			O_0			O_{-1}			O_{-2}		
Digit	3			7			6			2			2		
Binary	011			111			110			010			010		
Binary digit	0	1	1	1	1	1	1	1	0	0	1	0	0	1	0

Hexadecimal ←→ Binary (cont'd)

- Binary to hexadecimal
  - Group digits by **four** from right to left
  - Proceeding from the left and to the right
  - Replace each four-bit group with the equivalent hexadecimal digit
- Hexadecimal to binary
  - Replace each hexadecimal digit by four bits, binary

Hexadecimal ←→ Binary: Sample

- $(A80C1)_{16} = (?)_2$
- $(11001010111001100011101)_2 = (?)_{16}$

Hexadecimal ←→ Binary: Sample (cont'd)

- $(A80C1)_{16} = (?)_2$
  - $(A80C1)_{16} = (1010\ 1000\ 0000\ 1100\ 0001)$
  - $(A80C1)_{16} = (10101000000011000001)_2$

- $(11001010111001100011101)_2 = (?)_{16}$
  - $(11001010111001100011101)_2 = (11\ 0010\ 1011\ 1001\ 1000\ 1101)$
  - $(11001010111001100011101)_2 = (0011\ 0010\ 1011\ 1001\ 1000\ 1101)$
  - $(11001010111001100011101)_2 = (3\ 2\ B\ 9\ 8\ D)_{16}$

Hexadecimal ←→ Octal

- Octal to hexadecimal
  - Two steps, using binary as intermediate base
  1) Converting octal to binary
     - Replace each octal digits with 3 bits
  2) Converting binary to hexadecimal dig
     - Regroup binary digits into 4-bit hexadecimal digits

- Hexadecimal to octal
  1) Converting hexadecimal to binary
     - Replace each hexadecimal digits with 4 bits
  2) Converting binary to hexadecimal dig
     - Regroup binary digits into 3-bit octal digits

Hexadecimal ←→ Octal: Sample

- $(2304)_8 = (?)_{16}$
- $(4FA01)_{16} = (?)_8$

Hexadecimal ←→ Octal: Sample

- $(2304)_8 = (?)_{16}$
  - $(2304)_8 = (010\ 011\ 000\ 100)_2$
  - $(2304)_8 = (010011000100)_2$
  - $(2304)_8 = (0100\ 1100\ 0100)_2$
  - $(2304)_8 = (4\ C\ 4)_{16}$

Hexadecimal ←→ Octal (cont'd)

- $(4FA01)_{16} = (?)_8$
  - $(4FA01)_{16} = (0100\ 1111\ 1010\ 0000\ 0001)_2$
  - $(4FA01)_{16} = (01001111101000000001)_2$
  - $(4FA01)_{16} = (01\ 001\ 111\ 101\ 000\ 000\ 001)_2$
  - $(4FA01)_{16} = (001\ 001\ 111\ 101\ 000\ 000\ 001)_2$
  - $(4FA01)_{16} = (1\ 1\ 7\ 5\ 0\ 0\ 1)_8$

Arithmetic Operations

Single-bit Binary Addition

- Consider two binary digits (A, B)
- How can we add these two binary digits?

$$\begin{array}{rr} & A \\ + & B \\ \hline C & S \end{array}$$

Single-bit Binary Addition (cont'd)

- Add two binary digits (A, B)

```
   A
+  B
-----
   C  S
```

```
   0        0        1        1
+  0     +  1     +  0     +  1
-----    -----    -----    -----
   0  0     0  1     0  1     1  0
```

Single-bit Binary Addition (cont'd)

- Consider two binary digits (A, B) and a carry bit C_i
- How can we add these two binary digits with carry bits?

$$\begin{array}{rr} & C_i \\ & A \\ + & B \\ \hline C & S \end{array}$$

Single-bit Binary Addition (cont'd)

- Add three binary digits (A, B, C_i)
- Consider C_i is 0

C_i	0	0	0
A	0	0	1
+ B	+ 0	+ 1	+ 0
C S	0 0	0 1	0 1

0
1
+ 1
1 0

Single-bit Binary Addition (cont'd)

- Add three binary digits (A, B, C_i)
- Consider C_i is 1

C_i	1	1	1
A	0	0	1
+ B	+ 0	+ 1	+ 0
C S	0 1	1 0	1 0

1	1
1	1
+ 1	+ 1
1 1	1 1

Multi-bit Binary Addition

- Consider two binary numbers (A, B) and a carry bit C_i
- How can we add these two binary numbers with a carry bit?

$$
\begin{array}{rr}
 & C_i \\
 & A \\
+ & B \\
\hline
C & S
\end{array}
$$

Multi-bit Binary Addition (cont'd)

- Add two binary numbers (A, B) and a carry bit C_i

$$\begin{array}{rr} & C_i \\ & A \\ + & B \\ \hline C & S \end{array}$$

$$\begin{array}{rr} & 0 \\ & 01100 \\ + & 10001 \\ \hline 0 & 11101 \end{array} \qquad \begin{array}{rr} & 0 \\ & 11101 \\ + & 10001 \\ \hline 1 & 01110 \end{array}$$

Multi-bit Binary Addition: Samples

- Add two binary numbers (A, B) and a carry bit C_i

	C_i
	A
+	B
C	S

								0
	1	1	0	1	1	0	0	0
+	1	0	0	0	1	1	0	1

								0
	1	1	0	1	1	1	1	1
+	1	0	0	0	1	0	0	0

Multi-bit Binary Addition: Sample 1


```
                0
    1 1 0 1 1 0 0 0
+   1 0 0 0 1 1 0 1
-----------------------
                1
```

Multi-bit Binary Addition: Sample 1


```
          0                    0
    1 1 0 1 1 0 0 0      1 1 0 1 1 0 0 0
+   1 0 0 0 1 1 0 1  +   1 0 0 0 1 1 0 1
-----------------    -----------------
          1                  0 1
```

Multi-bit Binary Addition: Sample 1

0	0	0
1 1 0 1 1 0 0 0	1 1 0 1 1 0 0 0	1 1 0 1 1 0 0 0
+ 1 0 0 0 1 1 0 1	+ 1 0 0 0 1 1 0 1	+ 1 0 0 0 1 1 0 1
1	0 1	1 0 1

Multi-bit Binary Addition: Sample 1


```
          0                 0                 0
  1 1 0 1 1 0 0 0   1 1 0 1 1 0 0 0   1 1 0 1 1 0 0 0
+ 1 0 0 0 1 1 0 1 + 1 0 0 0 1 1 0 1 + 1 0 0 0 1 1 0 1
-----------------   -----------------   -----------------
        1                 0 1               1 0 1
```

```
    1     0
  1 1 0 1 1 0 0 0
+ 1 0 0 0 1 1 0 1
-----------------
      0 1 0 1
```

Multi-bit Binary Addition: Sample 1

```
          0
  1 1 0 1 1 0 0 0
+ 1 0 0 0 1 1 0 1
-----------------
          1
```

```
          0
  1 1 0 1 1 0 0 0
+ 1 0 0 0 1 1 0 1
-----------------
        0 1
```

```
          0
  1 1 0 1 1 0 0 0
+ 1 0 0 0 1 1 0 1
-----------------
      1 0 1
```

```
        1   0
  1 1 0 1 1 0 0 0
+ 1 0 0 0 1 1 0 1
-----------------
      0 1 0 1
```

```
        1 1   0
  1 1 0 1 1 0 0 0
+ 1 0 0 0 1 1 0 1
-----------------
    0 0 1 0 1
```

Multi-bit Binary Addition: Sample 1


```
          0          0          0
  1 1 0 1 1 0 0 0  1 1 0 1 1 0 0 0  1 1 0 1 1 0 0 0
+ 1 0 0 0 1 1 0 1 + 1 0 0 0 1 1 0 1 + 1 0 0 0 1 1 0 1
-----------------  -----------------  -----------------
          1            0 1              1 0 1

        1   0        1 1   0          1 1   0
  1 1 0 1 1 0 0 0  1 1 0 1 1 0 0 0  1 1 0 1 1 0 0 0
+ 1 0 0 0 1 1 0 1 + 1 0 0 0 1 1 0 1 + 1 0 0 0 1 1 0 1
-----------------  -----------------  -----------------
        0 1 0 1          0 0 1 0 1        1 0 0 1 0 1
```

Multi-bit Binary Addition: Sample 1


```
          0
  1101100 0
+ 1000110 1
-----------
          1
```

```
          0
  1101100 0
+ 1000110 1
-----------
        0 1
```

```
          0
  1101100 0
+ 1000110 1
-----------
       10 1
```

```
    1     0
  1101100 0
+ 1000110 1
-----------
       010 1
```

```
    1 1   0
  1101100 0
+ 1000110 1
-----------
      0010 1
```

```
    1 1   0
  1101100 0
+ 1000110 1
-----------
     10010 1
```

```
    1 1   0
  1101100 0
+ 1000110 1
-----------
    110010 1
```

Multi-bit Binary Addition: Sample 1


```
          0                 0                 0
  1 1 0 1 1 0 0 0   1 1 0 1 1 0 0 0   1 1 0 1 1 0 0 0
+ 1 0 0 0 1 1 0 1 + 1 0 0 0 1 1 0 1 + 1 0 0 0 1 1 0 1
-----------------   -----------------   -----------------
        1                 0 1               1 0 1

    1   0             1 1   0           1 1   0
  1 1 0 1 1 0 0 0   1 1 0 1 1 0 0 0   1 1 0 1 1 0 0 0
+ 1 0 0 0 1 1 0 1 + 1 0 0 0 1 1 0 1 + 1 0 0 0 1 1 0 1
-----------------   -----------------   -----------------
      0 1 0 1           0 0 1 0 1         1 0 0 1 0 1

    1 1   0           1 1   0
  1 1 0 1 1 0 0 0   1 1 0 1 1 0 0 0
+ 1 0 0 0 1 1 0 1 + 1 0 0 0 1 1 0 1
-----------------   -----------------
    1 1 0 0 1 0 1   1 0 1 1 0 0 1 0 1
```

Multi-bit Binary Addition: Sample 2


```
          0                  0                  0
  1 1 0 1 1 1 1 1    1 1 0 1 1 1 1 1    1 1 0 1 1 1 1 1
+ 1 0 0 0 1 0 0 0  + 1 0 0 0 1 0 0 0  + 1 0 0 0 1 0 0 0
-----------------  -----------------  -----------------
        1                  1 1                1 1 1

      1   0              1 1   0            1 1   0
  1 1 0 1 1 1 1 1    1 1 0 1 1 1 1 1    1 1 0 1 1 1 1 1
+ 1 0 0 0 1 0 0 0  + 1 0 0 0 1 0 0 0  + 1 0 0 0 1 0 0 0
-----------------  -----------------  -----------------
      0 1 1 1            0 0 1 1 1          1 0 0 1 1 1

    1 1   0            1 1   0
  1 1 0 1 1 1 1 1    1 1 0 1 1 1 1 1
+ 1 0 0 0 1 0 0 0  + 1 0 0 0 1 0 0 0
-----------------  -----------------
    1 1 0 0 1 1 1      1 0 1 1 0 0 1 1 1
```

Thank You