Part 1

FROM ZERO TO ONE

***Digital Design and Computer Architecture*, 2nd Edition**

David Money Harris and Sarah L. Harris

BBM 231

Hours: Mondays: **13:00 – 16:00**

Instructors:

- Section 1: Ufuk Çelikcan
- Section 2: Özgür Erkent
- Section 3: Murat Aydos
- **Register the course on Piazza:**

https://piazza.com/hacettepe.edu.tr/fall2024/bbm231

BBM 233 Lab

- We will announce the details later
- Follow the announcements on Piazza
- https://piazza.com/hacettepe.edu.tr/fall2024/bbm233
- You will be using online circuit simulators and verilog software to do the lab.

BBM231 Grading

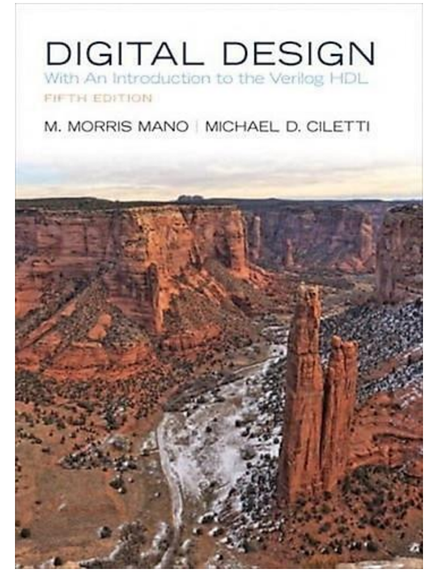
- Class Participation: 5%
- 1 Midterm exam: 35%
- Final exam: 40%
- Online Quizzes: 20%

Quizzes

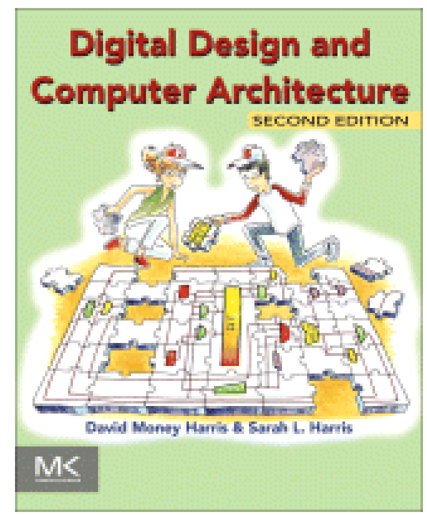
- Will be administered online via Hadi(probably)
- 9(?) Quizzes

References

- M. Morris Mano, Digital Design: With an Introduction to the Verilog HDL, **<u>5th Edition</u>**, Prentice Hall, 2013



- Digital Design and Computer Architecture 2nd edition, Harris and Harris



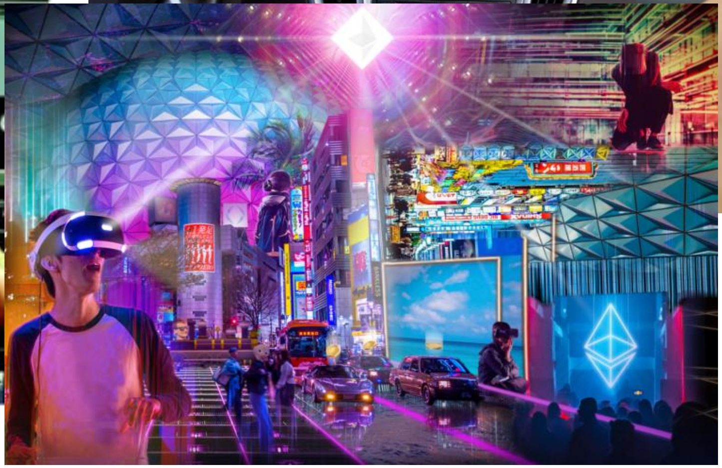
- Other references
  - Tons of digital design books
  - Lectures from MIT Open Courseware and Stanford

Chapter 1 :: Topics

- **Background**
- **The Game Plan**
- **The Art of Managing Complexity**
- **The Digital Abstraction**
- **Number Systems**
- **Logic Gates**
- **Logic Levels**
- **CMOS Transistors**
- **Power Consumption**

Background

- Microprocessors have revolutionized our world
  - Smartphones, Internet, rapid advances in medicine, etc.
- The semiconductor industry has grown from $21 billion in 1985 to $300 billion in 2011



The Game Plan

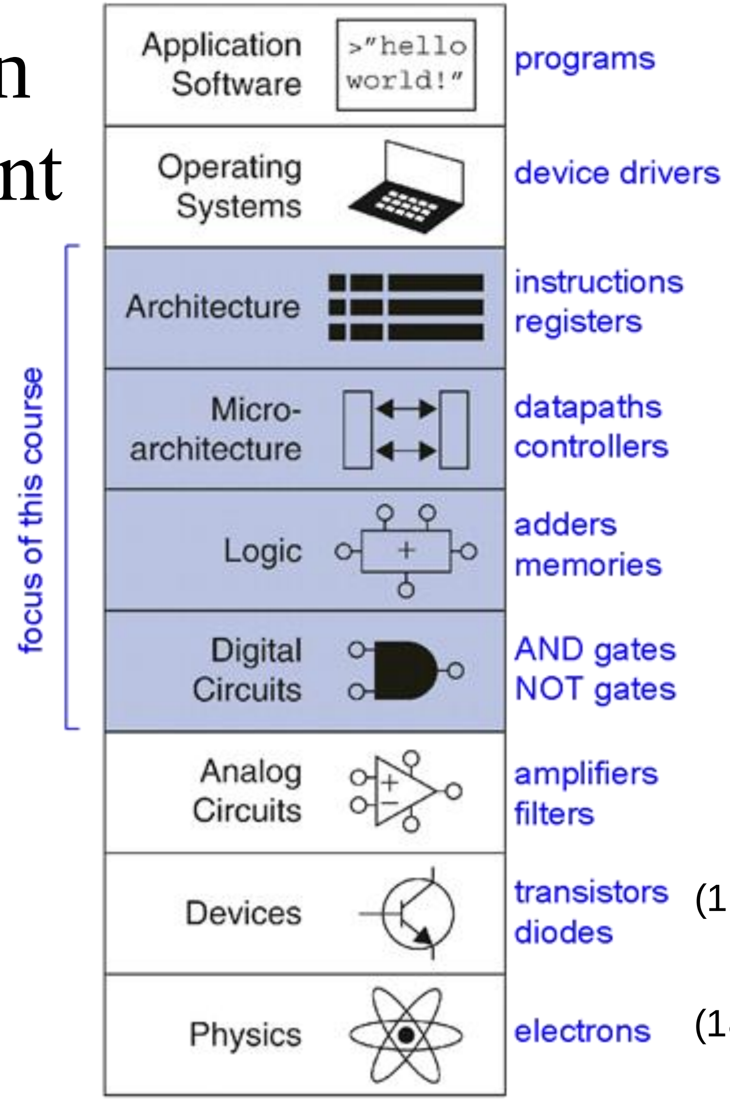
- Purpose of course:
  - Understand what's under the hood of a computer
  - Learn the principles of digital design
  - Learn to systematically debug increasingly complex designs

The Art of Managing Complexity

- Abstraction
- Discipline
- The Three –Y’s
  - Hierarchy
  - Modularity
  - Regularity

Abstraction

- Hiding details when they aren't important



(1947)

(1897)

Discipline

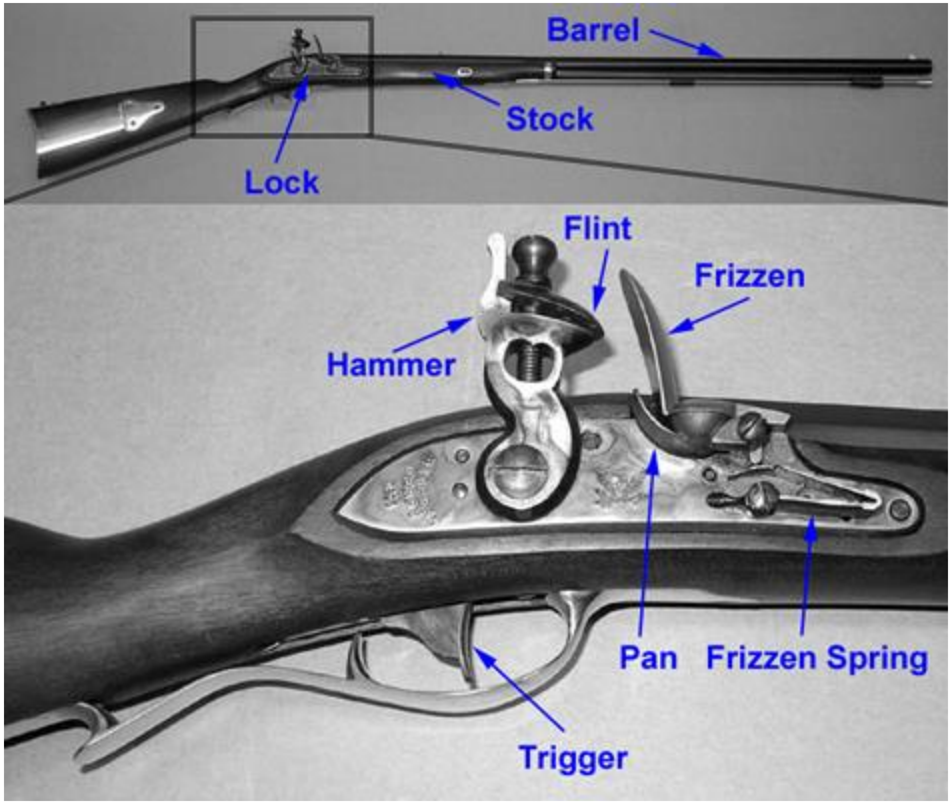
- Intentionally restrict design choices
- Example: Digital discipline
  - Discrete voltages instead of continuous
  - Simpler to design than analog circuits – can build more sophisticated systems
  - Digital systems replacing analog predecessors:
    - i.e., digital cameras, digital television, cell phones, CDs

The Three -Y's

- **Hierarchy**
  - A system divided into modules and submodules
- **Modularity**
  - Having well-defined functions and interfaces
- **Regularity**
  - Encouraging uniformity, so modules can be easily reused

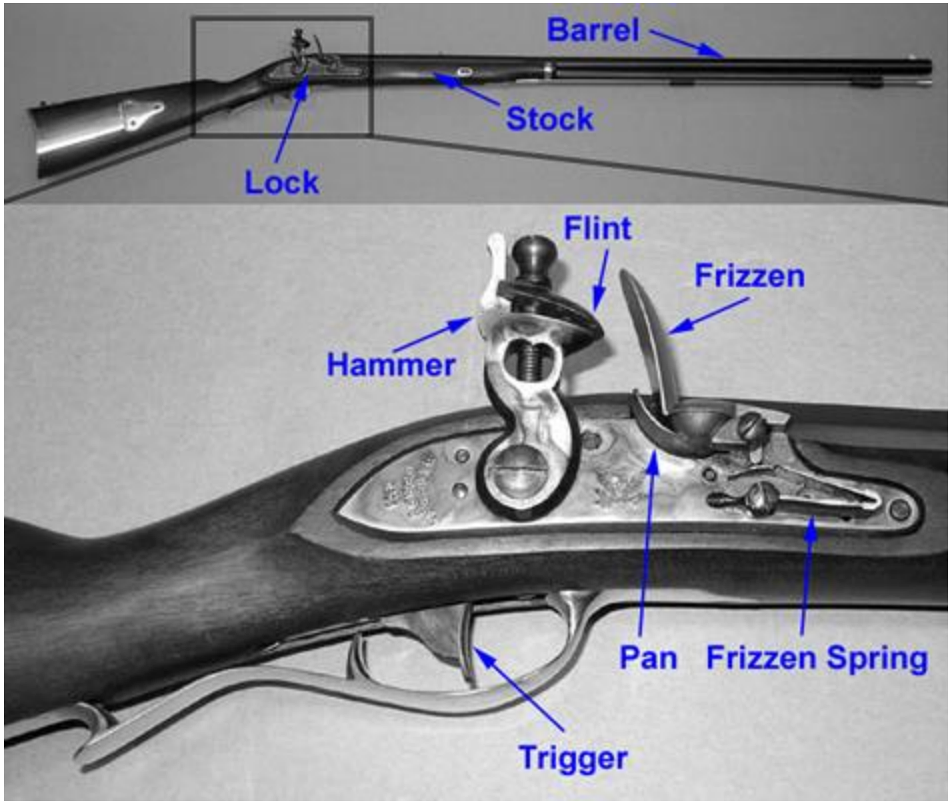
Example: The Flintlock Rifle

- **Hierarchy**
  - **Three main modules:** lock, stock, and barrel
  - **Submodules of lock:** hammer, flint, frizzen, etc.



Example: The Flintlock Rifle

- **Modularity**
  - **Function of stock:** mount barrel and lock
  - **Interface of stock:** length and location of mounting pins
- **Regularity**
  - Interchangeable parts

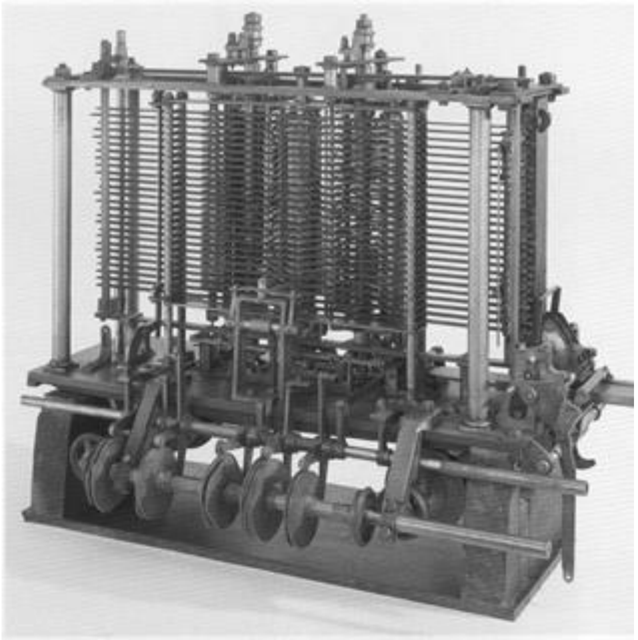


The Digital Abstraction

- Most physical variables are **continuous**
  - Voltage on a wire
  - Frequency of an oscillation
  - Position of a mass
- Digital abstraction considers **discrete subset** of values

The Analytical Engine

- Designed by Charles Babbage from 1834 – 1871
- Considered to be the first digital computer
- Built from mechanical gears, where each gear represented a discrete value (0-9)
- Babbage died before it was finished



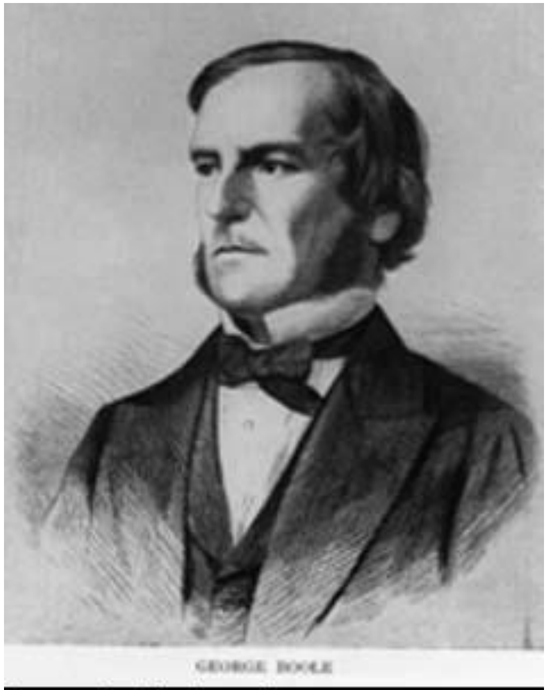
Digital Discipline: Binary Values

- **Two discrete values:**
  - 1’s and 0’s
  - 1, TRUE, HIGH
  - 0, FALSE, LOW
- **1 and 0:** voltage levels, rotating gears, fluid levels, etc.
- Digital circuits use **voltage** levels to represent 1 and 0
- ***Bit***: *B*inary dig*it*

George Boole, 1815-1864

FROM ZERO TO ONE

- Born to working class parents
- Taught himself mathematics and joined the faculty of Queen’s College in Ireland.
- Wrote *An Investigation of the Laws of Thought* (1854)
- Introduced **binary variables**
- Introduced the three fundamental logic operations: **AND, OR, and NOT**.



Scanned at the American Institute of Physics

Number Systems

- Decimal numbers

1000's column
100's column
10's column
1's column

$5374_{10} =$

- Binary numbers

8's column
4's column
2's column
1's column

$1101_2 =$

Number Systems

- Decimal numbers

1000's column
100's column
10's column
1's column

$$5374_{10} = \underbrace{5 \times 10^3}_{\text{five thousands}} + \underbrace{3 \times 10^2}_{\text{three hundreds}} + \underbrace{7 \times 10^1}_{\text{seven tens}} + \underbrace{4 \times 10^0}_{\text{four ones}}$$

- Binary numbers

8's column
4's column
2's column
1's column

$$1101_2 = \underbrace{1 \times 2^3}_{\text{one eight}} + \underbrace{1 \times 2^2}_{\text{one four}} + \underbrace{0 \times 2^1}_{\text{no two}} + \underbrace{1 \times 2^0}_{\text{one one}} = 13_{10}$$

Powers of Two

- $2^0 =$
- $2^1 =$
- $2^2 =$
- $2^3 =$
- $2^4 =$
- $2^5 =$
- $2^6 =$
- $2^7 =$
- $2^8 =$
- $2^9 =$
- $2^{10} =$
- $2^{11} =$
- $2^{12} =$
- $2^{13} =$
- $2^{14} =$
- $2^{15} =$

Powers of Two

- $2^0 = 1$
- $2^1 = 2$
- $2^2 = 4$
- $2^3 = 8$
- $2^4 = 16$
- $2^5 = 32$
- $2^6 = 64$
- $2^7 = 128$
- $2^8 = 256$
- $2^9 = 512$
- $2^{10} = 1024$
- $2^{11} = 2048$
- $2^{12} = 4096$
- $2^{13} = 8192$
- $2^{14} = 16384$
- $2^{15} = 32768$
- Handy to memorize up to 2^9

Number Conversion

- Binary to decimal conversion:
  - Convert 10011_2 to decimal

- Decimal to binary conversion:
  - Convert 47_{10} to binary

Number Conversion

- Binary to decimal conversion:
  - Convert 10011_2 to decimal
  - $16\times1 + 8\times0 + 4\times0 + 2\times1 + 1\times1 = 19_{10}$

- Decimal to binary conversion:
  - Convert 47_{10} to binary
  - $32\times1 + 16\times0 + 8\times1 + 4\times1 + 2\times1 + 1\times1 = 101111_2$

FROM ZERO TO ONE

Binary Values and Range

- *N*-digit decimal number
  – How many values?
  – Range?
  – Example: 3-digit decimal number:

- *N*-bit binary number
  – How many values?
  – Range:
  – Example: 3-digit binary number:

FROM ZERO TO ONE

Binary Values and Range

- *N*-digit decimal number
  - How many values? 10^N
  - Range? $[0, 10^N - 1]$
  - Example: 3-digit decimal number:
    - 10^3 = 1000 possible values
    - Range: [0, 999]
- *N*-bit binary number
  - How many values? 2^N
  - Range: $[0, 2^N - 1]$
  - Example: 3-digit binary number:
    - 2^3 = 8 possible values
    - Range: [0, 7] = $[000_2$ to $111_2]$

FROM ZERO TO ONE

ELSEVIER

Hexadecimal Numbers

Hex Digit	Decimal Equivalent	Binary Equivalent
0	0	
1	1	
2	2	
3	3	
4	4	
5	5	
6	6	
7	7	
8	8	
9	9	
A	10	
B	11	
C	12	
D	13	
E	14	
F	15	

Hexadecimal Numbers

Hex Digit	Decimal Equivalent	Binary Equivalent
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Hexadecimal Numbers

- Base 16
- Shorthand for binary

Hexadecimal to Binary

- Hexadecimal to binary conversion:
  - Convert $4AF_{16}$ (also written 0x4AF) to binary

- Hexadecimal to decimal conversion:
  - Convert 0x4AF to decimal

FROM ZERO TO ONE

Hexadecimal to Binary

- Hexadecimal to binary conversion:
  - Convert $4AF_{16}$ (also written 0x4AF) to binary
  - $0100\ 1010\ 1111_2$
- Hexadecimal to decimal conversion:
  - Convert $4AF_{16}$ to decimal
  - $16^2 \times 4 + 16^1 \times 10 + 16^0 \times 15 = 1199_{10}$

FROM ZERO TO ONE

Bits, Bytes, Nibbles...

- Bits

10010110

most significant bit

least significant bit

- Bytes & Nibbles

byte

10010110

nibble

- Bytes

CEBF9AD7

most significant byte

least significant byte

Large Powers of Two

- 2^{10} = 1 kilo ≈ 1000 (1024)
- 2^{20} = 1 mega ≈ 1 million (1,048,576)
- 2^{30} = 1 giga ≈ 1 billion (1,073,741,824)

Estimating Powers of Two

- What is the value of 2^{24}?

- How many values can a 32-bit variable represent?

Estimating Powers of Two

- What is the value of 2^{24}?

  $2^4 \times 2^{20} \approx$ **16 million**

- How many values can a 32-bit variable represent?

  $2^2 \times 2^{30} \approx$ **4 billion**

Addition

- Decimal

$$
\begin{array}{r}
\color{blue}{11}\ \ \ \ \ \ \color{blue}{\leftarrow \text{carries}} \\
3734\ \\
+\ \ 5168\ \\
\hline
8902\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
\end{array}
$$

- Binary

$$
\begin{array}{r}
\color{blue}{11}\ \ \ \ \ \ \color{blue}{\leftarrow \text{carries}} \\
1011\ \\
+\ \ 0011\ \\
\hline
1110\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
\end{array}
$$

FROM ZERO TO ONE

ELSEVIER

Binary Addition Examples

- Add the following 4-bit binary numbers

$$\begin{array}{r} 1001 \\ +\ 0101 \\ \hline \end{array}$$

- Add the following 4-bit binary numbers

$$\begin{array}{r} 1011 \\ +\ 0110 \\ \hline \end{array}$$

Binary Addition Examples

- Add the following 4-bit binary numbers

$$\begin{array}{r} {\color{blue}1} \\ 1001 \\ +\ \ 0101 \\ \hline {\color{blue}1110} \end{array}$$

- Add the following 4-bit binary numbers

$$\begin{array}{r} {\color{blue}111} \\ 1011 \\ +\ \ 0110 \\ \hline {\color{blue}10001} \end{array}$$

Overflow!

Overflow

- Digital systems operate on a **fixed number of bits**
- Overflow: when result is too big to fit in the available number of bits
- See previous example of 11 + 6

FROM ZERO TO ONE

ELSEVIER

Signed Binary Numbers

- Sign/Magnitude Numbers
- Two's Complement Numbers

Sign/Magnitude Numbers

- 1 sign bit, N-1 magnitude bits
- Sign bit is the most significant (left-most) bit
  - Positive number: sign bit = 0
  - Negative number: sign bit = 1

$$A : \{a_{N-1}, a_{N-2}, \cdots a_2, a_1, a_0\}$$

$$A = (-1)^{a_{n-1}} \sum_{i=0}^{n-2} a_i 2^i$$

- Example, 4-bit sign/mag representations of ± 6:

  +6 =

  - 6 =

- Range of an N-bit sign/magnitude number:

FROM ZERO TO ONE

ELSEVIER

Sign/Magnitude Numbers

- 1 sign bit, N-1 magnitude bits
- Sign bit is the most significant (left-most) bit
  - Positive number: sign bit = 0
  - Negative number: sign bit = 1

$$A:\{a_{N-1},a_{N-2},\cdots a_2,a_1,a_0\}$$

$$A=(-1)^{a_{n-1}}\sum_{i=0}^{n-2} a_i 2^i$$

- Example, 4-bit sign/mag representations of ± 6:

  +6 = **0110**

  - 6 = **1110**

- Range of an N-bit sign/magnitude number:

  $[-(2^{N-1}-1), 2^{N-1}-1]$

Sign/Magnitude Numbers

- Problems:
  - Addition doesn't work, for example -6 + 6:

```
   1110
 + 0110
 ------
  10100 (wrong!)
```

  - Two representations of 0 (± 0):

```
   1000
   0000
```

Two's Complement Numbers

- Don't have same problems as sign/magnitude numbers:
  - Addition works
  - Single representation for 0

Two's Complement Numbers

- Msb has value of -2^{N-1}

$$A = a_{n-1}\left(-2^{n-1}\right) + \sum_{i=0}^{n-2} a_i 2^i$$

- Most positive 4-bit number:
- Most negative 4-bit number:
- The most significant bit still indicates the sign (1 = negative, 0 = positive)
- Range of an *N*-bit two's comp number:

Two's Complement Numbers

- Msb has value of -2^{N-1}

$$A = a_{n-1}\left(-2^{n-1}\right) + \sum_{i=0}^{n-2} a_i 2^i$$

- Most positive 4-bit number: **0111**
- Most negative 4-bit number: **1000**
- The most significant bit still indicates the sign (1 = negative, 0 = positive)
- Range of an *N*-bit two's comp number:

$[-(2^{N-1}), 2^{N-1}-1]$

"Taking the Two's Complement"

- Flip the sign of a two's complement number
- Method:
    1. Invert the bits
    2. Add 1
- Example: Flip the sign of $3_{10} = 0011_2$

“Taking the Two’s Complement”

- Flip the sign of a two’s complement number
- Method:
    1. Invert the bits
    2. Add 1
- Example: Flip the sign of $3_{10} = 0011_2$
    1. **1100**
    2. **+ 1**

    1101 = -3$_{10}$

Two's Complement Examples

- Take the two's complement of $6_{10} = 0110_2$

- What is the decimal value of 1001_2?

FROM ZERO TO ONE

ELSEVIER

Two's Complement Examples

- Take the two's complement of $6_{10} = 0110_2$
  1. 1001
  2. + 1

     $1010_2 = -6_{10}$

- What is the decimal value of the two's complement number 1001_2?
  1. 0110
  2. + 1

     $0111_2 = 7_{10}$, so $1001_2 = -7_{10}$

Two's Complement Addition

- Add 6 + (-6) using two's complement numbers

```
      0110
  +   1010
  ________
```

- Add -2 + 3 using two's complement numbers

```
      1110
  +   0011
  ________
```

Two’s Complement Addition

- Add 6 + (-6) using two’s complement numbers

$$\begin{array}{r} \color{blue}{111} \\ 0110 \\ +\ \ 1010 \\ \hline 10000 \end{array}$$

- Add -2 + 3 using two’s complement numbers

$$\begin{array}{r} \color{blue}{111} \\ 1110 \\ +\ \ 0011 \\ \hline 10001 \end{array}$$

Two's Complement Addition

- What is the 2's complement of 0000 ?

Two's Complement Addition

- What is the 2's complement of 0000 ?

$$\begin{array}{r} 1111 \\ +\quad\quad 1 \\ \hline 10000 \end{array}$$

>> Also 0000, single representation for 0

Increasing Bit Width

- **Extend number from *N* to *M* bits (*M* > *N*) :**
  - Sign-extension
  - Zero-extension

FROM ZERO TO ONE

Sign-Extension

- Sign bit copied to msb's
- Number value is same

- **Example 1:**
  - 4-bit representation of 3 = 0011
  - 8-bit sign-extended value: 00000011
- **Example 2:**
  - 4-bit representation of -5 = 1011
  - 8-bit sign-extended value: 11111011

ELSEVIER

Zero-Extension

FROM ZERO TO ONE

- Zeros copied to msb's
- For unsigned numbers
  - Value changes for negative signed numbers

- **Example 1:**
  - 4-bit value = $0011_2 = 3_{10}$
  - 8-bit zero-extended value: $00000011 = 3_{10}$
- **Example 2:**
  - 4-bit value = $1011 = -5_{10}$
  - 8-bit zero-extended value: $00001011 = 11_{10}$

Number System Comparison

Number System	Range
Unsigned	$[0, 2^N-1]$
Sign/Magnitude	$[-(2^{N-1}-1), 2^{N-1}-1]$
Two's Complement	$[-2^{N-1}, 2^{N-1}-1]$

For example, 4-bit representation:

	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Unsigned									0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Two's Complement	1000	1001	1010	1011	1100	1101	1110	1111	0000	0001	0010	0011	0100	0101	0110	0111								
Sign/Magnitude		1111	1110	1101	1100	1011	1010	1001	0000 1000	0001	0010	0011	0100	0101	0110	0111								

FROM ZERO TO ONE