Credit: Some slides are from the “Invitation to Computer Science” book by G. M. Schneider, J. L. Gersting and some from the “Digital Design” book by M. M. Mano and M. D. Ciletti.
WHAT IS COMPUTER SCIENCE?
What is Computer Science?

COMMON MISCONCEPTIONS (1)

- “Computer science is the study of computers”
- Incomplete – theoretical work began (1920-1940) before computers
- CS became an independent field of study late 1950’s, early 1960’s
- Theoretical CS – relies on formal models rather than “real” machines
- CS “is no more about computers than astronomy is about telescopes”, etc.

From “Invitation to Computer Science”
What is Computer Science?

COMMON MISCONCEPTIONS (2)

- “Computer science is the study of how to write computer programs”

- Programming is important, but it is just a tool for studying new ideas, representing information or testing the solution to a problem.

- A Program is a means to an end, not the end itself.

- E.g. Searching a list such as the NYC phone Directory

From “Invitation to Computer Science”
What is Computer Science?

COMMON MISCONCEPTIONS (3)

- “Computer science is the study of the uses and applications of computers and software”

- E.g. word processors, databases, spreadsheets, etc.

- Many people USE software, but the Computer Scientist is responsible for specifying, designing, building and testing software packages and the systems on which they run.

From “Invitation to Computer Science”
What is Computer Science?

All of the following concepts are incomplete and do not capture the richness and diversity of this exciting field:

- computers,
- programming languages,
- software applications, and uses.
Computer Science is the study of algorithms (= methods)

including:

1. their formal and mathematical properties
2. their hardware realizations
3. their linguistic realizations
4. their applications

From “Invitation to Computer Science”
Measure the height of a tall building with a barometer

What would be your answer?

One student answered:

- “I would tie the barometer to a rope, hang it down from the top of the building to the bottom and measure the length of the rope!”

- Of course, the instructor rejects the answer since it doesn’t include any “physics”

Check the following for two different versions of the `legend’:
Where does the word ‘algorithm’ come from?

From a Persian mathematician, astronomer and geographer: Mohammed ibn-Musa al-Khwarizmi

“Algorithmi” is the latin form of his name

He contributed to science by

- Decimal positional number system (e.g., $32 = 10^1 \times 3 + 10^0 \times 2$)
- Presented the first systematic solutions to linear and quadratic equations

In fact, the word “Algebra” comes from one of his operators (al-jabr: subtracting a number from both sides of an equation) for solving equations

What does ‘algorithm’ mean?

- “A procedure or formula for solving a problem”
- “A set of instructions to be followed to solve a problem”
- “an effective method expressed as a finite list of well-defined instructions for calculating a function”
- “step-by-step procedure for calculations”
What is an algorithm?

An algorithm is a list that looks like

- STEP 1: Do something
- STEP 2: Do something
- STEP 3: Do something
- ...
- ...
- ...
- ...
- STEP N: Stop, you are finished

From “Invitation to Computer Science”
Valid Operations in Algorithms

■ **Sequential** – simple well-defined task, usually declarative sentence.

■ **Conditional** - “ask a question and select the next operation on the basis of the answer to the question – usually an “if-then” or “if then else”

■ **Iterative** - “looping” instructions – repeat a set of instructions

From “Invitation to Computer Science”
An example algorithm

Algorithm for Adding Two $m$-Digit Numbers

**Given:** $m \geq 1$ and two positive numbers each containing $m$ digits, $a_{m-1} a_{m-2} \ldots a_0$ and $b_{m-1} b_{m-2} \ldots b_0$

**Wanted:** $c_m c_{m-1} c_{m-2} \ldots c_0$, where $c_m c_{m-1} c_{m-2} \ldots c_0 = (a_{m-1} a_{m-2} \ldots a_0) + (b_{m-1} b_{m-2} \ldots b_0)$

**Algorithm:**

1. **Step 1** Set the value of *carry* to 0.
2. **Step 2** Set the value of *i* to 0.
3. **Step 3** While the value of *i* is less than or equal to $m - 1$, repeat the instructions in steps 4 through 6.
4. **Step 4** Add the two digits $a_i$ and $b_i$ to the current value of *carry* to get $c_i$.
5. **Step 5** If $c_i \geq 10$, then reset $c_i$ to $(c_i - 10)$ and reset the value of *carry* to 1; otherwise, set the new value of *carry* to 0.
6. **Step 6** Add 1 to *i*, effectively moving one column to the left.
7. **Step 7** Set $c_m$ to the value of *carry*.
8. **Step 8** Print out the final answer, $c_m c_{m-1} c_{m-2} \ldots c_0$.
9. **Step 9** Stop.

From “Invitation to Computer Science”
Why are algorithms important?

- If we can specify an algorithm to solve a problem then we can automate its solution.

- No algorithm => No software => No automation!
Can we find algorithms to all problems?

NO!

- There are problems which have no generalized solutions – unsolvable or intractable
- Some with an algorithm would take so long to execute that the algorithm is useless
- Some problems we have not yet discovered an algorithm for

From “Invitation to Computer Science”
A formal definition of algorithm

“Starting from an initial state and initial input (perhaps empty), the instructions describe a computation that, when executed, will proceed through a finite number of well-defined successive states, eventually producing "output" and terminating at a final ending state.”
“Computation”

- Digital vs. analog computation
- Sequential vs. parallel computation
- Batch vs. interactive computation
- Evolutionary, molecular, quantum computation

“Physical computation” / “Digital Physics”
- ‘The whole universe is itself a computation’
Problem: Find temperature of the water if A&B were mixed together.

Any suggestions on how to solve it?
Computation in our brain

- Highly-connected network of neurons.
- How many neurons?
  - Approx. $10^{11}$ neurons and $10^{14}$ synapses.
- How do they transmit information?
  - Using nothing else than charged molecules.
Computation in our brain (cont’d)

- Each neuron gets input and produces an output using an “activation function”
Turing Machine

Von Neumann Architecture

DIGITAL COMPUTATION
The Harvard Mark-I

Grace M. Hopper working on the Harvard Mark-I, developed by IBM and Howard Aiken. The Mark-I remained in use at Harvard until 1959, even though other machines had surpassed it in performance, providing vital calculations for the navy in World War II.
Programming the ENIAC
History of Computation

- Read the reading material on this subject!
  - 15th of October: quiz from the reading material
Turing Machine

Von Neumann Architecture

DIGITAL COMPUTATION
A computer

Devices

Gates

Transistors
Everything in a PC is Binary
... well, almost ...

<table>
<thead>
<tr>
<th>States of a Bit</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 + 2 = 5</td>
<td>OFF</td>
<td>HIGH VOLTAGE</td>
</tr>
<tr>
<td>FALSE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 + 2 = 4</td>
<td>ON</td>
<td></td>
</tr>
<tr>
<td>TRUE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
A transistor

This circuit functions as a switch. In other words, based on the control voltage, the circuit either passes Vin to output or not.
A computer

Devices

Gates

Transistors
NOT Gate

\[
\begin{array}{c|c}
X & \overline{X} \\
0 & 1 \\
1 & 0 \\
\end{array}
\]
AND gate

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>X·Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Previously on CENG 111!
OR Gate

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>X+Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Today

- A digital circuit for an example problem
- N-bit addition using a digital circuit
Reminder

- Submit your lab schedules!
An example problem: Water Tank

<table>
<thead>
<tr>
<th>HI</th>
<th>LO</th>
<th>Pump</th>
<th>Drain</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

Schematic Representation

Truth Table Representation
Boolean Logic/Algebra

Pump = HI’.LO
Drain = HI.LO’

Boolean formula describing the circuit.
The binary addition

\[
\begin{array}{cccc}
0 & 1 & 0 & 1 \\
+ 0 & + 0 & + 1 & + 1 \\
\hline
0 & 1 & 1 & 0 \\
\end{array}
\]

Question (Binary notation) : \( 111010 + 11011 = ? \)
1-bit Half-adder

<table>
<thead>
<tr>
<th>Ai</th>
<th>Bi</th>
<th>Sum</th>
<th>Carry</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
1-bit full-adder

\[
\begin{array}{c}
A \\
B \\
CIn \\
S \\
CO
\end{array}
\begin{array}{cc}
0 & 0 \\
0 & 1 \\
1 & 0 \\
1 & 1 \\
0 & 0 \\
0 & 1 \\
1 & 0 \\
1 & 1 \\
\end{array}
\begin{array}{c}
A \\
B \\
CIn \\
S \\
CO
\end{array}
\begin{array}{cc}
0 & 0 \\
0 & 1 \\
1 & 0 \\
1 & 1 \\
0 & 0 \\
0 & 1 \\
1 & 0 \\
1 & 1 \\
\end{array}
\begin{array}{c}
A \\
B \\
CIn \\
S \\
CO
\end{array}
\begin{array}{cc}
0 & 0 \\
0 & 1 \\
1 & 0 \\
1 & 1 \\
0 & 0 \\
0 & 1 \\
1 & 0 \\
1 & 1 \\
\end{array}
\begin{array}{c}
A \\
B \\
CIn \\
S \\
CO
\end{array}
\begin{array}{cc}
0 & 0 \\
0 & 1 \\
1 & 0 \\
1 & 1 \\
0 & 0 \\
0 & 1 \\
1 & 0 \\
1 & 1 \\
\end{array}
\begin{array}{c}
A \\
B \\
CIn \\
S \\
CO
\end{array}
\begin{array}{cc}
0 & 0 \\
0 & 1 \\
1 & 0 \\
1 & 1 \\
0 & 0 \\
0 & 1 \\
1 & 0 \\
1 & 1 \\
\end{array}
\begin{array}{c}
A \\
B \\
CIn \\
S \\
CO
\end{array}
\begin{array}{cc}
0 & 0 \\
0 & 1 \\
1 & 0 \\
1 & 1 \\
0 & 0 \\
0 & 1 \\
1 & 0 \\
1 & 1 \\
\end{array}
N-bit Adder