What is a System?

- A system is a set of related components that works as a whole to achieve a goal.

- A system contains:
  - Inputs
  - Behavior
  - Outputs

- Behavior is a function that translates inputs to outputs
System Components

- Components are electronic blocks: analog, digital, and mixed signal
  - Analog system has values from a continuous set

![Graph of an analog signal with values ranging from -5 to +5 volts over time.](image)
System Components

• Components are electronic blocks: digital, analog, and mixed signal
  ‣ Digital system is a system in which signals have a finite number of discrete values
Design design is what runs the world

1. GPS receiver
   Matches position with customised version of Google’s road maps

2. Laser range finder
   Rotating sensor scans 180m distance through 360° to generate 3D map of surroundings

3. Video camera
   Identifies other road users, lane markers and traffic signals

4. Radar
   Located at front and rear, detect proximity of obstacles

Speed: Limited to 40km/h to help ensure safety

Engine: 180km-range electric motor – equivalent to one used by Fiat’s 500e

Inertial motion sensors determine velocity and direction

Radar
Computer & Digital Design

- Boolean algebra serves as the basis for computer logic design
- Transistors are the means to implement Boolean algebra in modern computer systems
- Basic Boolean algebra
  - Set of Elements: \{0,1\}
  - Set of Operations: \{., +, ¬\}
- Digital equivalence
  - Signals: High = 5V = 1; Low = 0V = 0
Boolean Algebra to Digital Design

Symbol/Gate

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Gate</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOT</td>
<td><img src="image" alt="NOT gate" /></td>
</tr>
<tr>
<td>OR</td>
<td><img src="image" alt="OR gate" /></td>
</tr>
<tr>
<td>AND</td>
<td><img src="image" alt="AND gate" /></td>
</tr>
</tbody>
</table>

Truth table

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>F</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>

Transistor circuit

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Circuit</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOT</td>
<td><img src="image" alt="NOT circuit" /></td>
</tr>
<tr>
<td>OR</td>
<td><img src="image" alt="OR circuit" /></td>
</tr>
<tr>
<td>AND</td>
<td><img src="image" alt="AND circuit" /></td>
</tr>
</tbody>
</table>
## Logic Gates

<table>
<thead>
<tr>
<th>Name</th>
<th>Symbol</th>
<th>Function</th>
<th>Truth Table</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AND</strong></td>
<td>![AND Symbol]</td>
<td>$X = A \cdot B$ or $X = AB$</td>
<td>![AND Truth Table]</td>
</tr>
<tr>
<td><strong>OR</strong></td>
<td>![OR Symbol]</td>
<td>$X = A + B$</td>
<td>![OR Truth Table]</td>
</tr>
<tr>
<td><strong>NOT</strong></td>
<td>![NOT Symbol]</td>
<td>$X = A'$</td>
<td>![NOT Truth Table]</td>
</tr>
<tr>
<td><strong>NAND</strong></td>
<td>![NAND Symbol]</td>
<td>$X = (AB)'$</td>
<td>![NAND Truth Table]</td>
</tr>
<tr>
<td><strong>NOR</strong></td>
<td>![NOR Symbol]</td>
<td>$X = (A + B)'$</td>
<td>![NOR Truth Table]</td>
</tr>
<tr>
<td><strong>XOR</strong></td>
<td>![XOR Symbol]</td>
<td>$X = A \oplus B$ or $X = A'B + AB'$</td>
<td>![XOR Truth Table]</td>
</tr>
<tr>
<td><strong>XNOR</strong></td>
<td>![XNOR Symbol]</td>
<td>$X = (A \oplus B)'$ or $X = A'B' + AB$</td>
<td>![XNOR Truth Table]</td>
</tr>
</tbody>
</table>
Boolean Algebra to Digital Design

• Addition

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Sum</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Carry</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

• Truth Table

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Cin</th>
<th>Sum</th>
<th>Cout</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Boolean Algebra to Digital Design

- Truth Table

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>Sum</th>
<th>Cout</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

- Equations

\[
\text{Sum} = \text{Cin} \cdot A' \cdot B' + B \cdot \text{Cin}' \cdot A' + A \cdot \text{Cin}' \cdot B' + A \cdot B \cdot \text{Cin}
\]

\[
\text{Cout} = A' \cdot B \cdot \text{Cin} + A \cdot B' \cdot \text{Cin} + A \cdot \text{Cin}' \cdot B + B \cdot \text{Cin} \cdot A
\]

\[
= A \cdot B + A \cdot \text{Cin} + B \cdot \text{Cin}
\]
Boolean Algebra to Digital Design

\[ Sum = Cin \cdot A' \cdot B' + B \cdot Cin' \cdot A' + A \cdot Cin' \cdot B' + A \cdot B \cdot Cin \]
\[ \text{Cout} = A' \cdot B \cdot \text{Cin} + A \cdot B' \cdot \text{Cin} + A \cdot \text{Cin}' \cdot B + B \cdot \text{Cin} \cdot A \]

\[ = A \cdot B + A \cdot \text{Cin} + B \cdot \text{Cin} \]
Advantages of Digital Systems

• Digital representation is very suited for both numerical and non-numerical information or data processing
  ‣ Numerical values: 0, 4, 5567, 24354543, -2
  ‣ Non-numerical values: There will be light! (Each letter in this sentence)
Advantages of Digital Systems

- Digital representation is very suited for both numerical and non-numerical information or data processing
- Easier to design, particularly the automation design and fabrication
- Lower cost
- Easier to duplicate similar circuits
- Higher noise immunity
Advantages of Digital Systems

• Easier to control by electronics: the finite number of values in a digital system can be represented by a vector of signals with just two values (binary signals)
  ‣ 2 is 0010 and 10 is 1010

• Device to process this type of signal is very simple
  ‣ Can be a switch: open/close

• Adjustable precision
Integrated Circuit Based Designs

- Complex digital integrated circuits (ICs) are manufactured with the advent of Microelectronics Technology
  - The number of components fitted into a standard size IC represents its integration scale, also called density

![Diagram showing the progression from SSI to ULSI with general purpose, functionality, and application-specific components.]
Integrated Circuit Based Designs

• IC density
  ‣ SSI – Small Scale Integration: has about 10 gates
  ‣ MSI – Medium Scale Integration: has more than 10 but less than 100 gates
**Integrated Circuit Based Designs**

- **IC density**
  - **LSI** – Large Scale Integration: has more than 100 gates
  - **VLSI** – Very Large Scale Integration: has contains more than 300000 components per chip
• **IC density**

  ‣ **ULSI** - Ultra Large Scale Integration: has more than 1500000 components per chip
Modern Computing Systems

• Our physical world is analog and computing is interacting with the physical world:
  ‣ So we need to convert physical signals to digital then back to analog to communicate with the real world
Challenges with Digital Systems

- Our physical world is analog and computing is interacting with the physical world:
  - So we need to convert physical signals to digital then back to analog to communicate with the real world
    - This increases the complexity of digital systems
    - It leads to less precision
    - And lower speed
- This is why we have this class to learn how to mitigate these disadvantages
System Design Parameters

• Specification of system is the description of its function and other characteristics required for it:
  ‣ Speed, cost and power
System Design Process

- Functional specification
  - What is the function performed by the object?
  - Constraints: How fast? How much area? How much cost?
  - Refine abstract functional blocks into more concrete realizations
System Design Process

- Functional specification
- Implementation
  \- Assemble primitives into more complex building blocks
  \- Composition via wiring
  \- Choose among alternatives to improve the design
System Design Process

- Functional specification
- Debugging
  - Faulty systems: design flaws, composition flaws, component flaws
  - Design to make debugging easier
  - Hypothesis formation and troubleshooting skills
System Design Process

• Functional specification
  › A fire sprinkler system should spray water if high heat is sensed and the system is set to enabled
System Design Process

- **Functional specification**
  - A fire sprinkler system should spray water if high heat is sensed and the system is set to enabled.

- **Implementation**
  - Let Boolean variable $H$ represent “high heat is sensed,” $E$ represent “enabled,” and $W$ represent “spraying water.” Then an equation is: $W = H \text{ AND } E$
System Design Process

- Functional specification
  - A car alarm should sound if the alarm is enabled, and either the car is shaken or the door is opened
System Design Process

- Functional specification
  - A car alarm should sound if the alarm is enabled, and either the car is shaken or the door is opened.

- Implementation
  - Let $E$ represent “alarm is enabled,” $S$ represent “car is shaken,” $O$ represent “door is opened,” and $A$ represent “alarm sounds.” Then an equation is: $A = E \text{ AND } (S \text{ OR } O)$. 
System Design Process

- **Functional specification**
  - A car alarm should sound if the alarm is enabled, and either the car is shaken or the door is opened.

- **Implementation**
  - Alternatively, assuming that our door sensor $O$ represents “door is closed” instead of open (meaning $O=1$ when the door is closed, $0$ when open), we obtain the following equation: $A = E \text{ AND } (S \text{ OR NOT}(O))$. 
Digital Design Methodology

levels of abstraction
architecture

behavioral perspective
algorithm
subtasks
data/moves and operations
truth tables, state graphs
transfer functions

logicsynthesis
electrical

logic (aka gate-level)

structural perspective
top blocks and I/O
subblocks
ALUs, registers, memories
gates, latches, flip-flops
transistors, wires

system

front-end design

behavioral perspective
physical perspective

back-end design

physical perspective

START

FINISH

standard cells, macrocells
placement and routing
floorplan, partitioning
chip or board

mask polygons, detailed layout
Past & Present Design Roles

- Competence of systems engineers
- Product idea
  - Evaluation of functional needs and specification
  - Algorithm design
- Architecture design
- Technology-specific implementation
- IC fabrication data

- Competence of VLSI designers
• Goal: Design a hardware unit to do GCD of two integer numbers
Next Class

- Digital System Design