VLSI Topics

Tinoosh Mohsenin, Chintan Ptel

CMPE 641

Today

- Administrative items
- Syllabus and course overview
- Digital signal processing and hardware design overview
  - Digital Signal processing and applications
  - Digital logics
  - ASIC, FPGA, programmable processors

Course Communication

- Email
  - Urgent announcements
- Web page
  - http://www.csee.umbc.edu/~tinoosh/cmpe641/
- Office hours
  - By appointment

Course Description

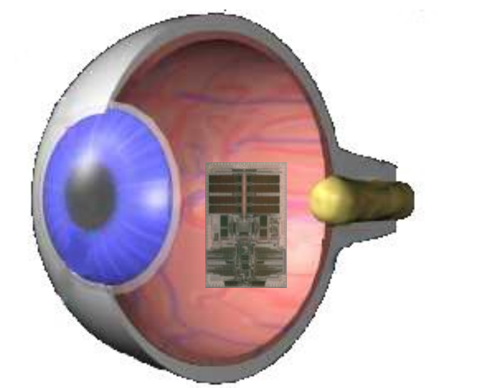
- Lectures
- Handouts
- Homework/ projects
  - Three/four HWs
- Midterm Exam
  - To be decided
- Final Project, Demonstration and Presentation

Lectures

- Ask questions at any time
- Participate in the class (%5 of your grade)
- Please silence phones
- Please hold conversations outside of class
- No computer usage in class

The Future: New Applications

- Very limited power budgets
- Require significant digital signal processing
- Must perform in real time
- Reconfigurable for different environments
- Require innovations in algorithm, architecture, and circuit design



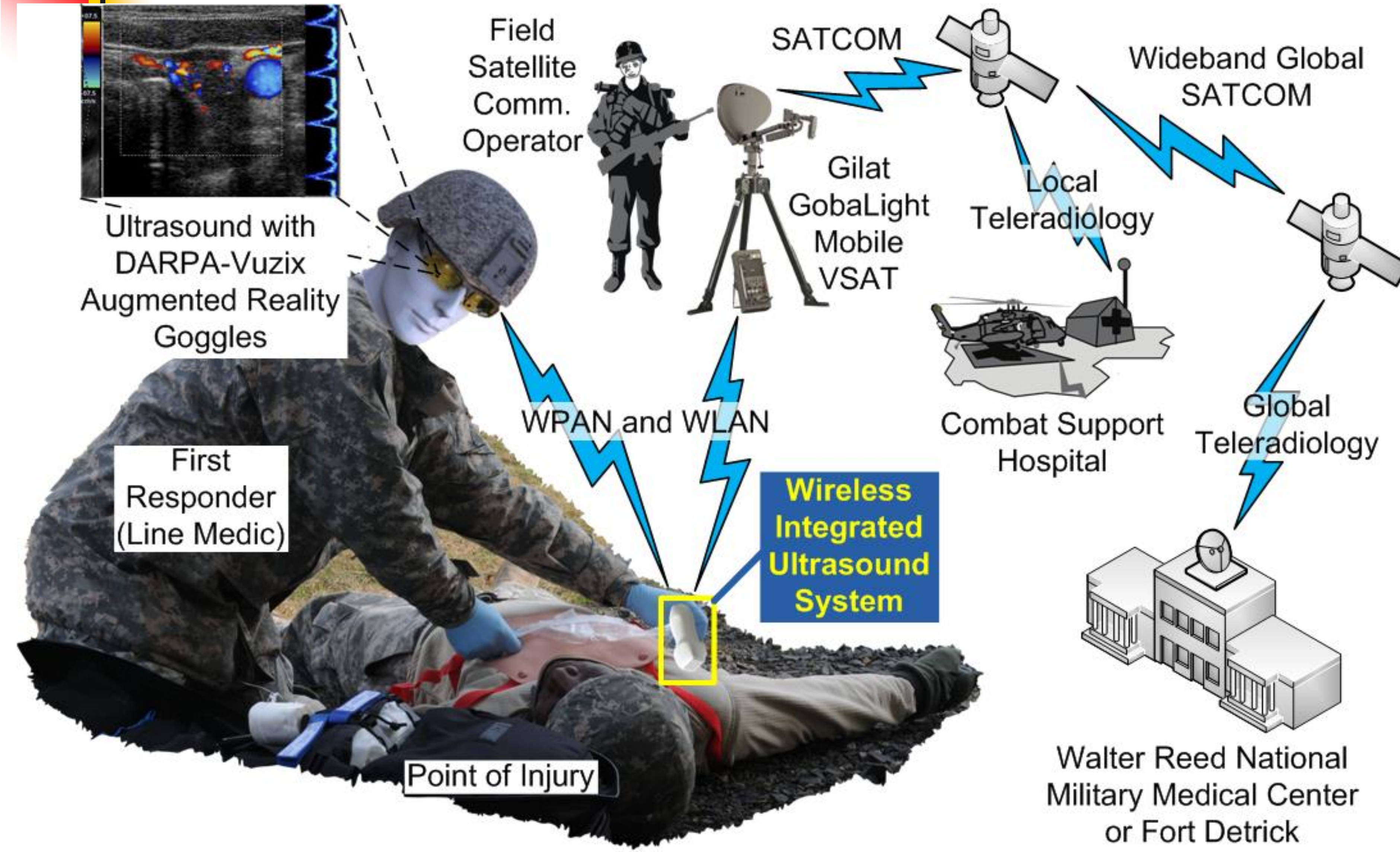
Smart Embedded Processing in Big Data World

- The vast quantities of real-time data produced by embedded sensors, smartphones and wearable systems present new challenges
  - Data transmission, storage, and analysis
  - Maintaining high throughput processing and low latency communications,
  - Low power consumption.
- Systems are getting smarter and independent
  - Incorporate adaptive and intelligent kernels to overcome the noise and false detection by combining the analysis of multi-modal signals.
- Reconfiguration and programmability are required to generalize hardware for different environments and tasks
  - Reduces design time and overall time to market
- Increasing energy-efficiency (i.e. ↑GOPS/W, ↓pJ/op) requires innovations in algorithms, programming models, processor architectures, and circuit design

Embedded Applications

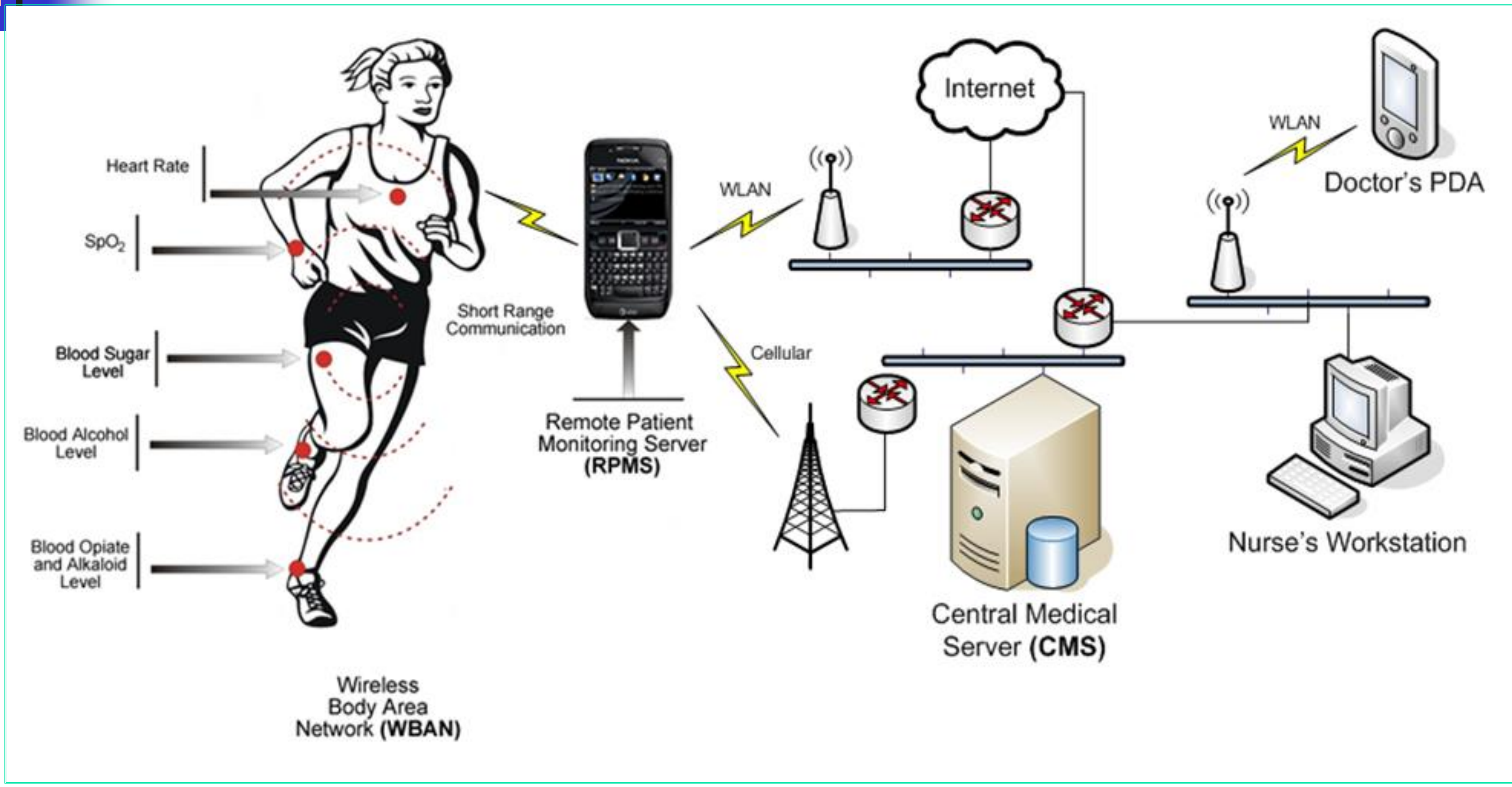
- Requirements:
  - Real time, low power, light weight, high accuracy
- Steps to design an embedded application on a programmable processor
  - Understand the target platform
    - e.g single processor vs multiprocessor
  - Understand the digital signal processing requirement for the application
    - What algorithms
    - How many data channels, how many bits per channel data
  - Break the application into multiple tasks
  - Write a code for each task and verify it using real/simulated data and examine the accuracy
  - Program the processor
    - Single core: all tasks in one core
    - Multi core: parallelize the tasks and program each core for the task

Future Military Applications



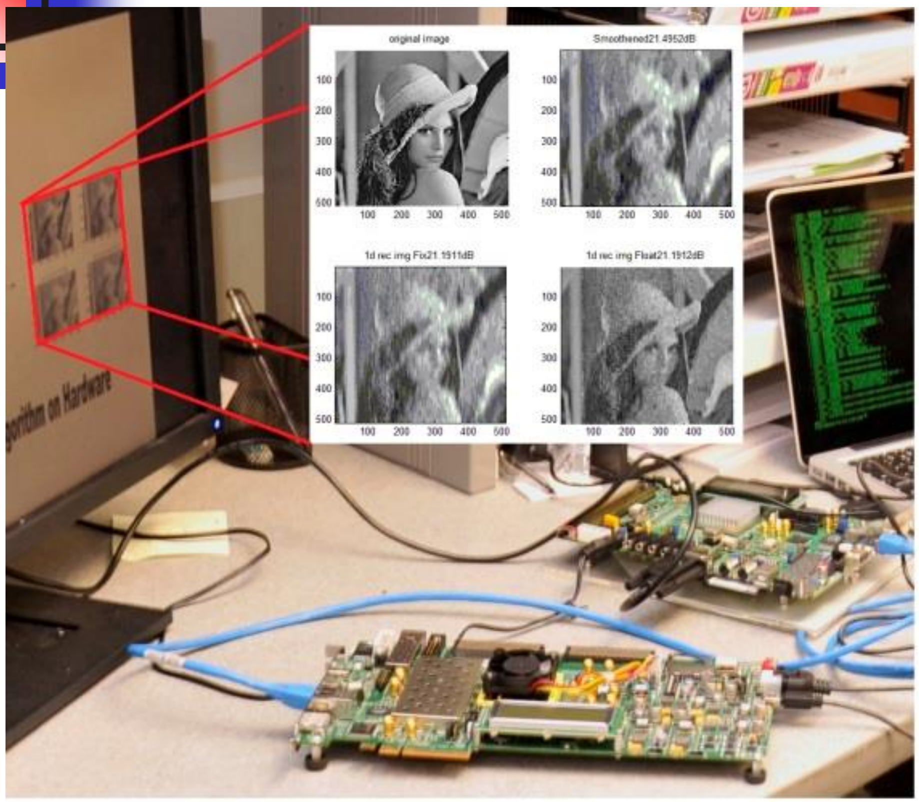
@ D. Truong

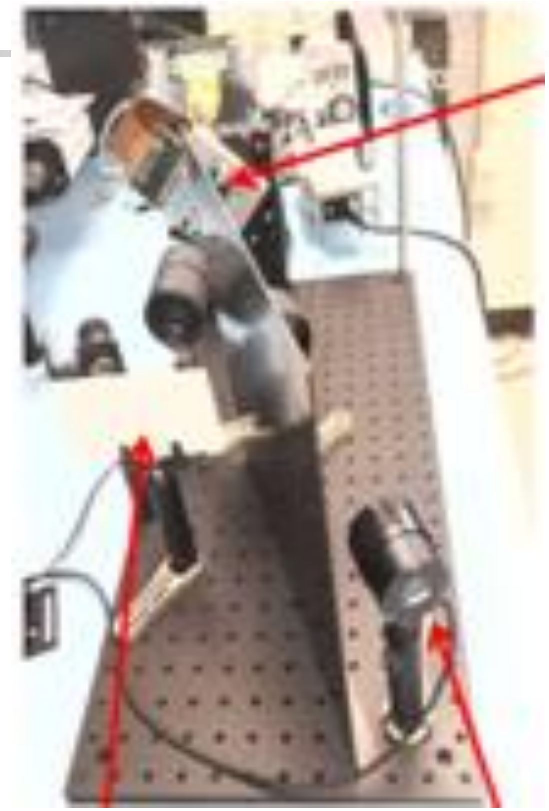
Wearable Medical Monitoring and Analysis



- Data must be acquired, analyzed and transmitted
- Some must be processed in real time
- Ultra low power processing

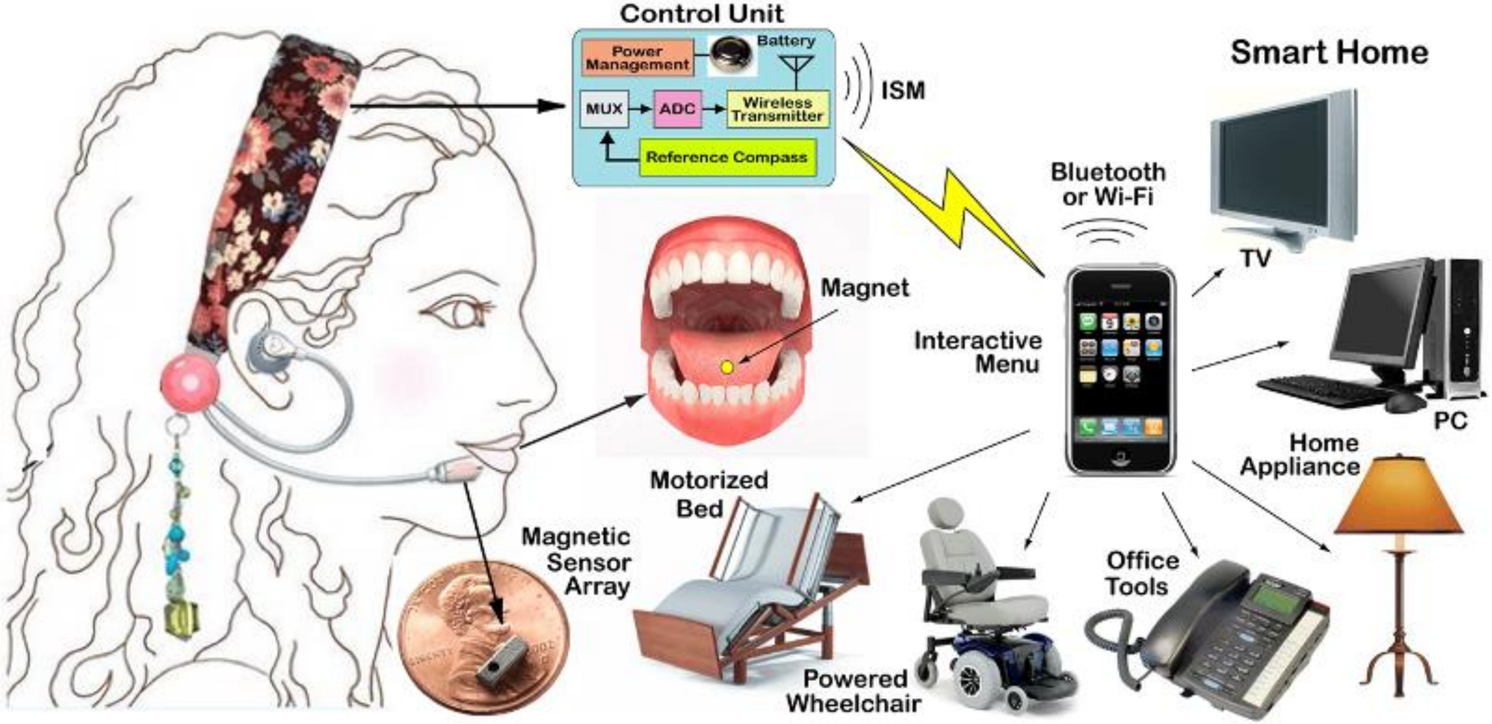
Compressive Sensing for Reduction in Data Transmission





- Single pixel camera setup at NASA Goddard
- Image reconstruction using compressive sensing on Virtex 7 FPGA

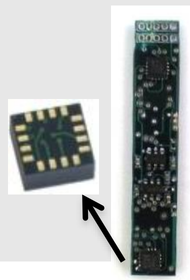
Tongue Drive System (TDS)

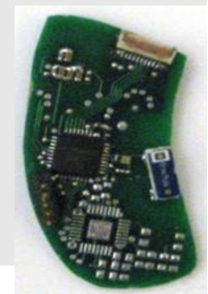


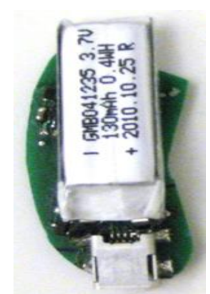
- A tongue-operated assistive technology that enables individuals with severe physical impairments to control their environments.
- An array of magnetic sensors detect the magnetic field variations resulted from the movements of a small magnetic tracer attached to the tongue, convert the sensed signals to the user commands in a local processor and wirelessly send the user command to the target device.

eTDS: Hardware

Headset Components	
1. Sensors: Four 3-axial magneto-resistive sensors (two on each pole)	**3. Control Unit:** MCU: TI CC2510 2.4 GHz RF Transceiver
2. Magnet: Disk-shaped [4.8mm × 1.5mm] Embedded in a titanium tongue stud	**4. Battery:** 130 mAh, 3.7 V, plus power management circuit



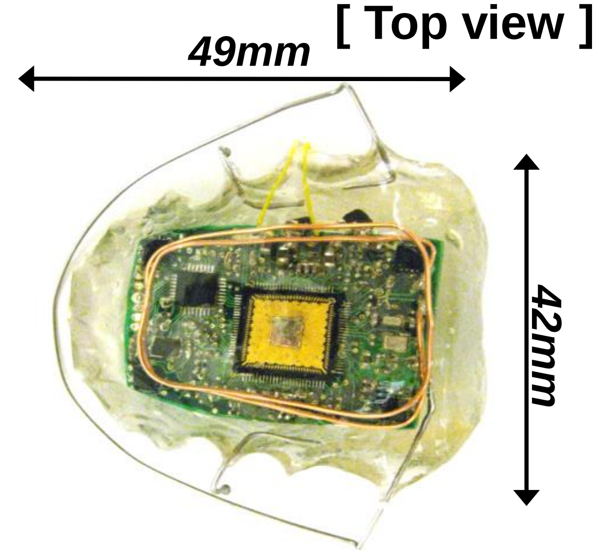


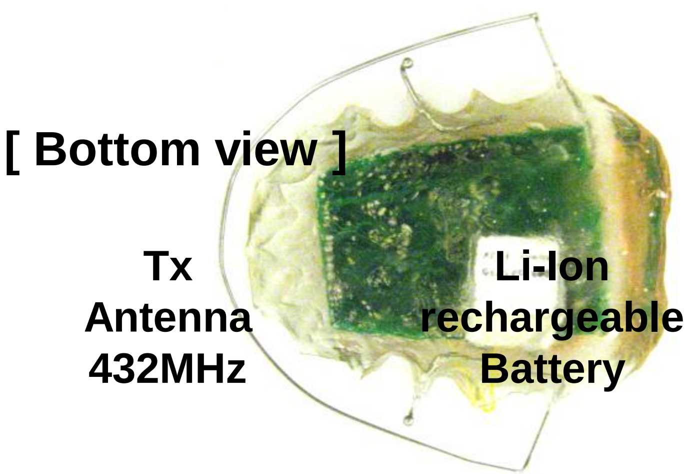


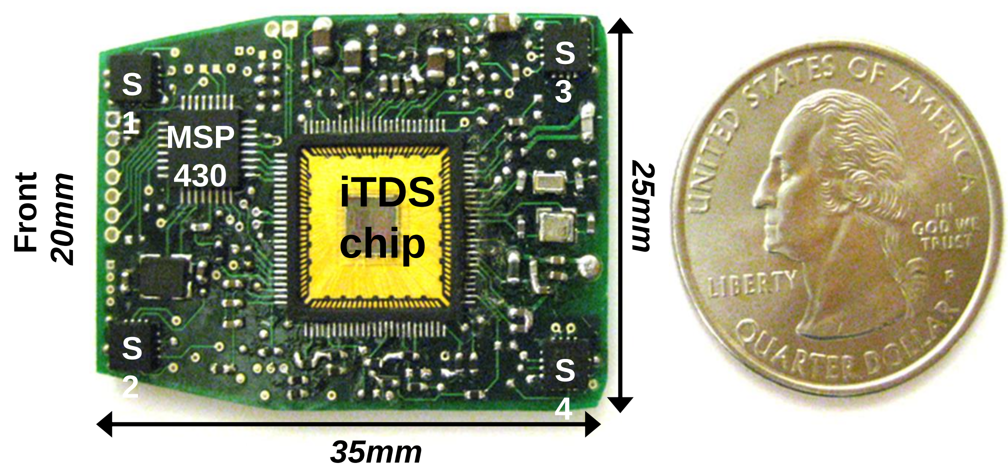
- eTDS has been clinically tested with NIH support at the top rehab institutes, such as Shepherd Center in Atlanta and Rehabilitation Institute of Chicago.

Current iTDS Prototype

- Transmits all the raw data to a computer to process
- High transmission volume cause high power consumption
  - Sends 20bits for each sensor at 50 Hz
  - There are 12 sensors => total is 12 Kbits/sec
- Size limitation restricts us to a 50mAh battery and consequently a shorter battery life







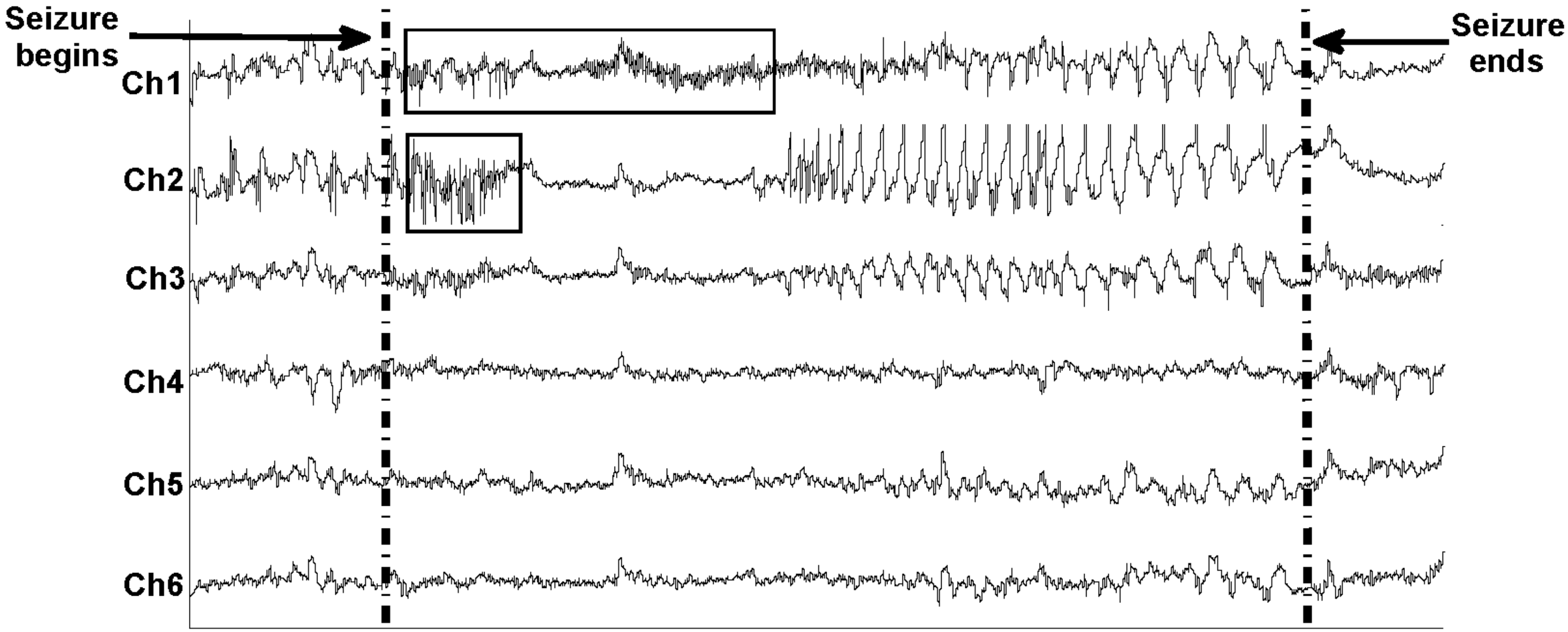
Wearable Seizure Detection



- Epilepsy is the 4th most common neurological disorder, 1 in 26 people may develop epilepsy in their lifetime.
- About 25% of epilepsy patients have intractable seizures which may occur with an unpredictable pattern, including during sleep when there may be less surveillance by family.
  - Places these patients at greatest risk from the potential morbidity and mortality of severe or sustained seizures.
- Current ambulatory seizure monitoring devices are infeasible for long-term and continuous use due to:
  - Large false positive/negative signals, noise due to patient activity, bulky equipment, high power consumption, and the inability of patients to carry on with their daily lives.

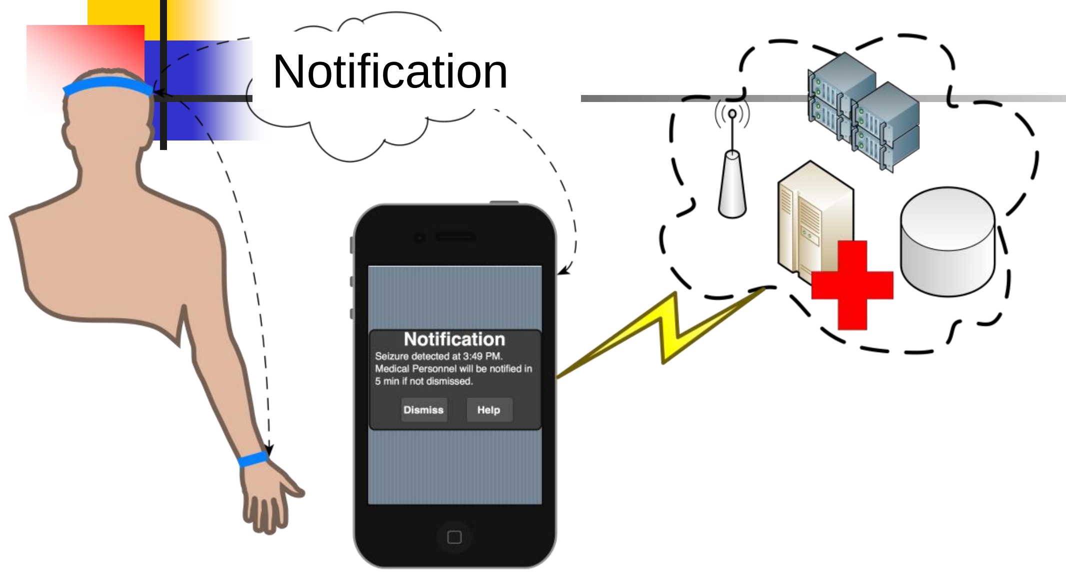
Seizure Detection Problem

- Electrical signals can be detected by EEG signals before or just at the start of clinical symptoms
  - The ability to detect can be used to warn the patient or alert caregiver
- Seizure patterns are unique to each patient and seizure and non-seizure EEG signals from the same patient can share similar characteristics
- Complex algorithms and multichannel detection is necessary for better detection

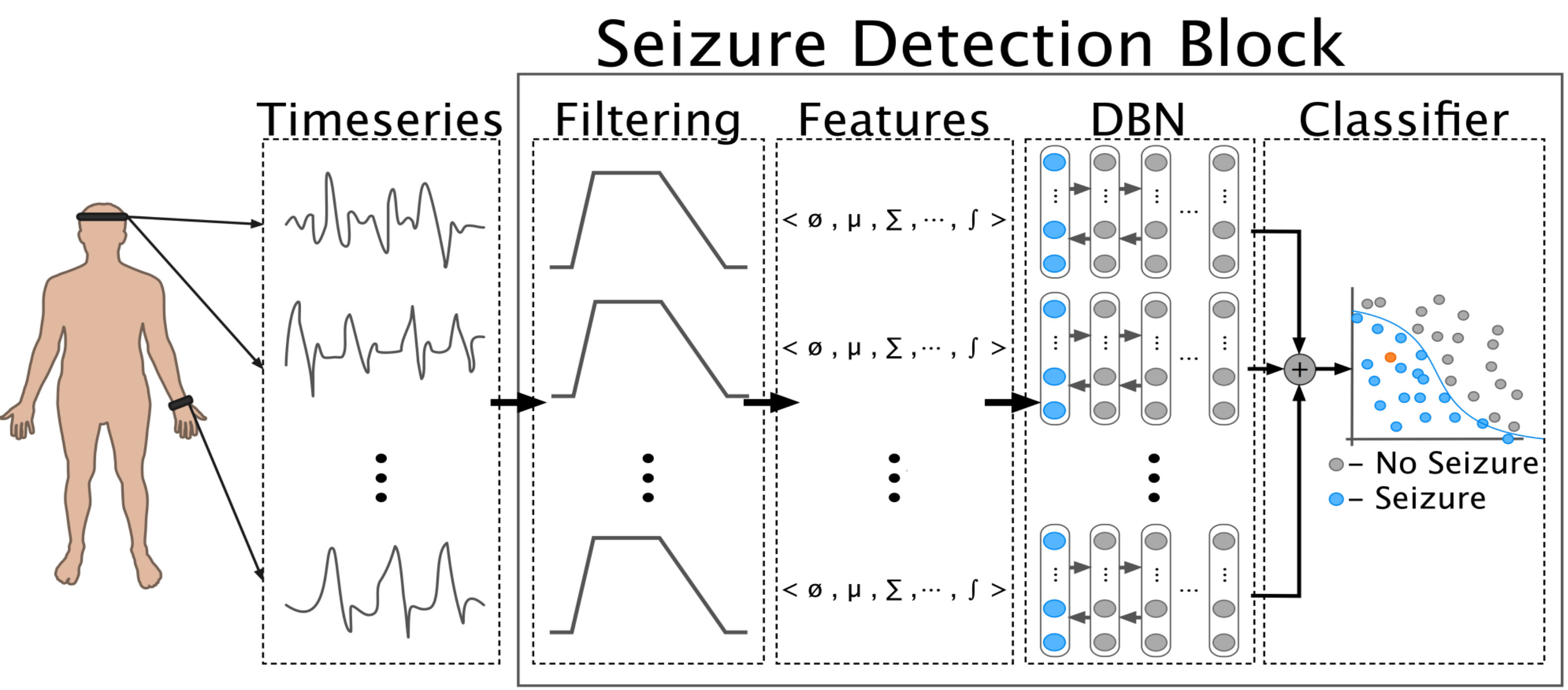


Chandler et al
BioCass 2011

A wearable solution for Multi-physiological signal processing



- **Headband sensors**
  EEG data, EOG, gyroscope data, and accelerometer
- **Wristband sensors**
  heart rate, blood flow, and blood oxygenation through pulse oximeter.



Digital Signal Processing vs Analog Processing

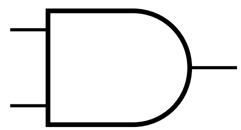
- DSP arithmetic is completely stable over process, temperature, and voltage variations
  - Ex: 2.0000 + 3.0000 = 5.0000 will always be true as long as the circuit is functioning correctly
- DSP energy-efficiencies are rapidly increasing
- Once a DSP processor has been designed in a portable format (gate netlist, HDL, software), very little effort is required to “port” (re-target) the design to a different processing technology. Analog circuits typically require a nearly-complete re-design.
- DSP capabilities are rapidly increasing
- Analog A/D speed x resolution product doubles every 5 years
- Digital processing performance doubles every 18-24 Months (6x to 10x every 5 years

Common Trends

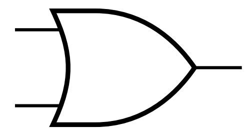
- Analog based →Digital based
  - Music: records, tapes → CDs
  - Video: VHS, 8mm → DVD, Blu-ray
  - Telephony, cell phones: analog (1G) → digital (2G, 3G, 4G, …)
  - Television: NTSC → digital (DVB, ATSC, ISDB, …)
  - Many new things use digital data and “speak” digital: computers, networks, digital appliances

Basic Digital Circuit Components

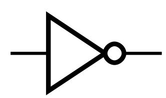
- Primitive components for logic design



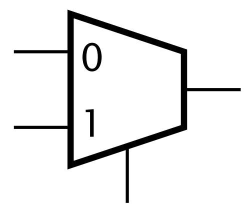
AND gate



OR gate



inverter



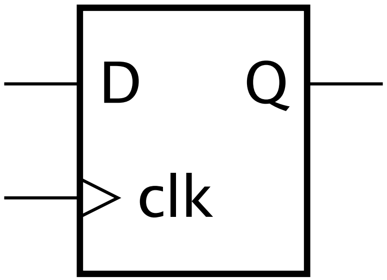
multiplexer

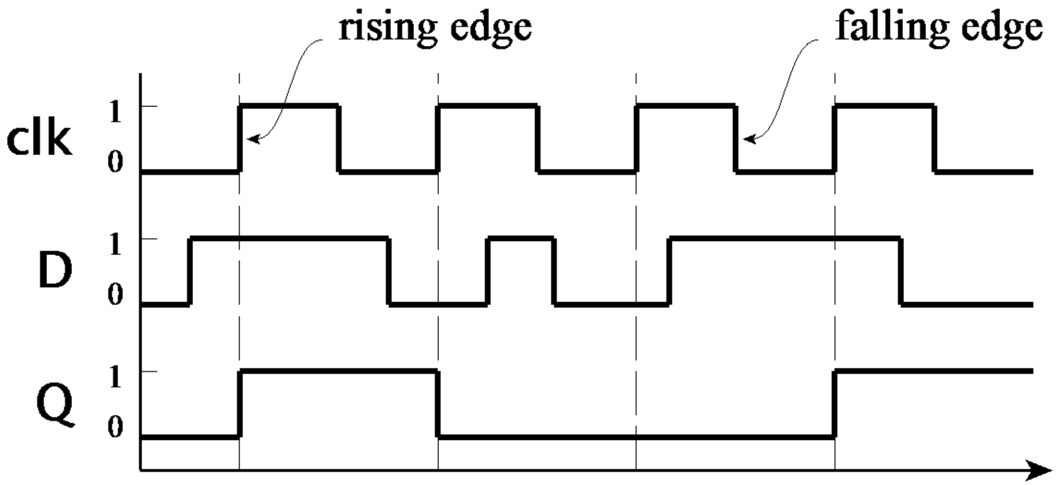
Sequential Circuits

- Circuit whose output values depend on current *and previous* input values
  - Include some form of storage of values
- Nearly all digital systems are sequential
  - Mixture of gates and storage components
  - Combinational parts transform inputs and stored values

Flipflops and Clocks

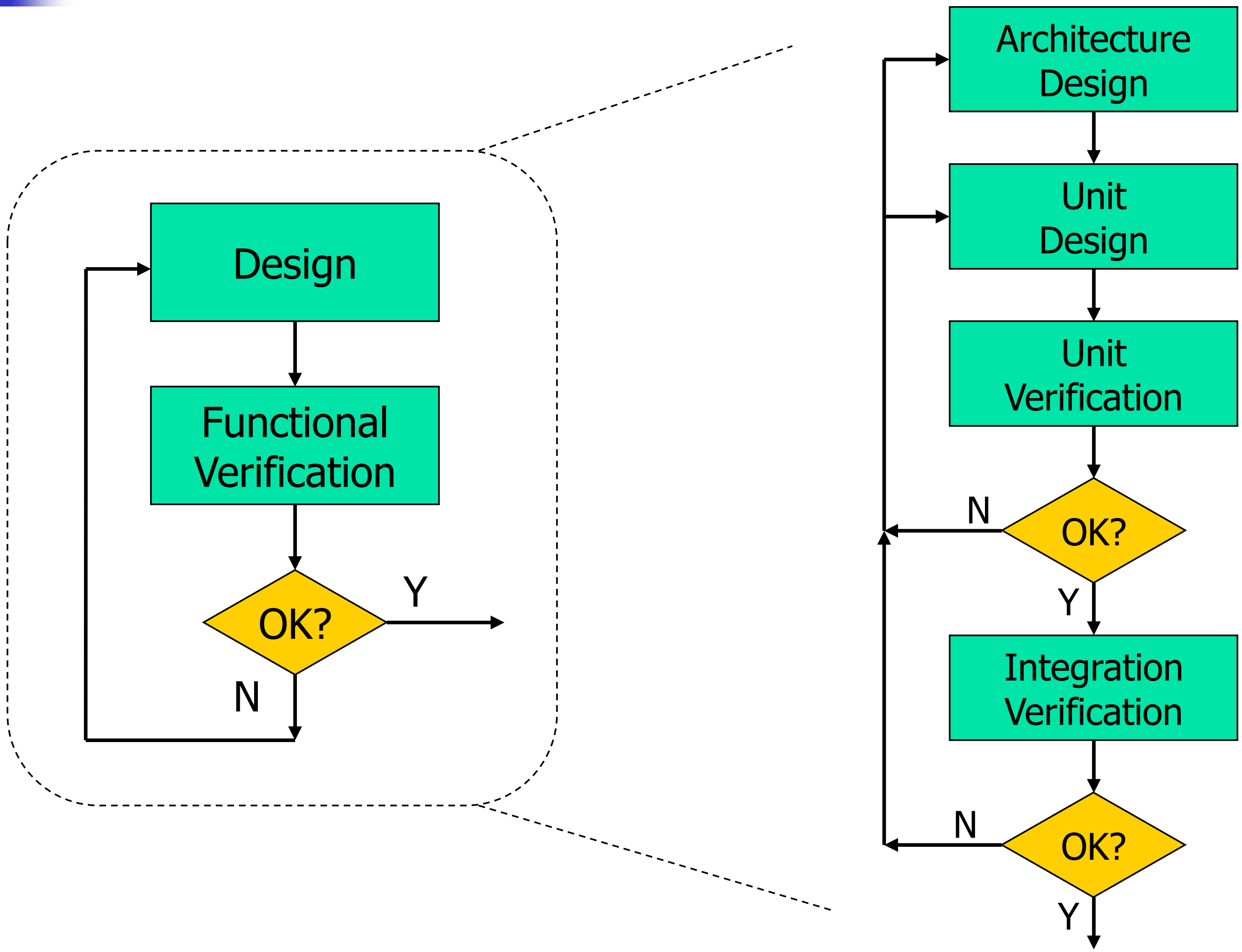
- Edge-triggered D-flipflop
  - stores one bit of information at a time





- Timing diagram
  - Graph of signal values versus time

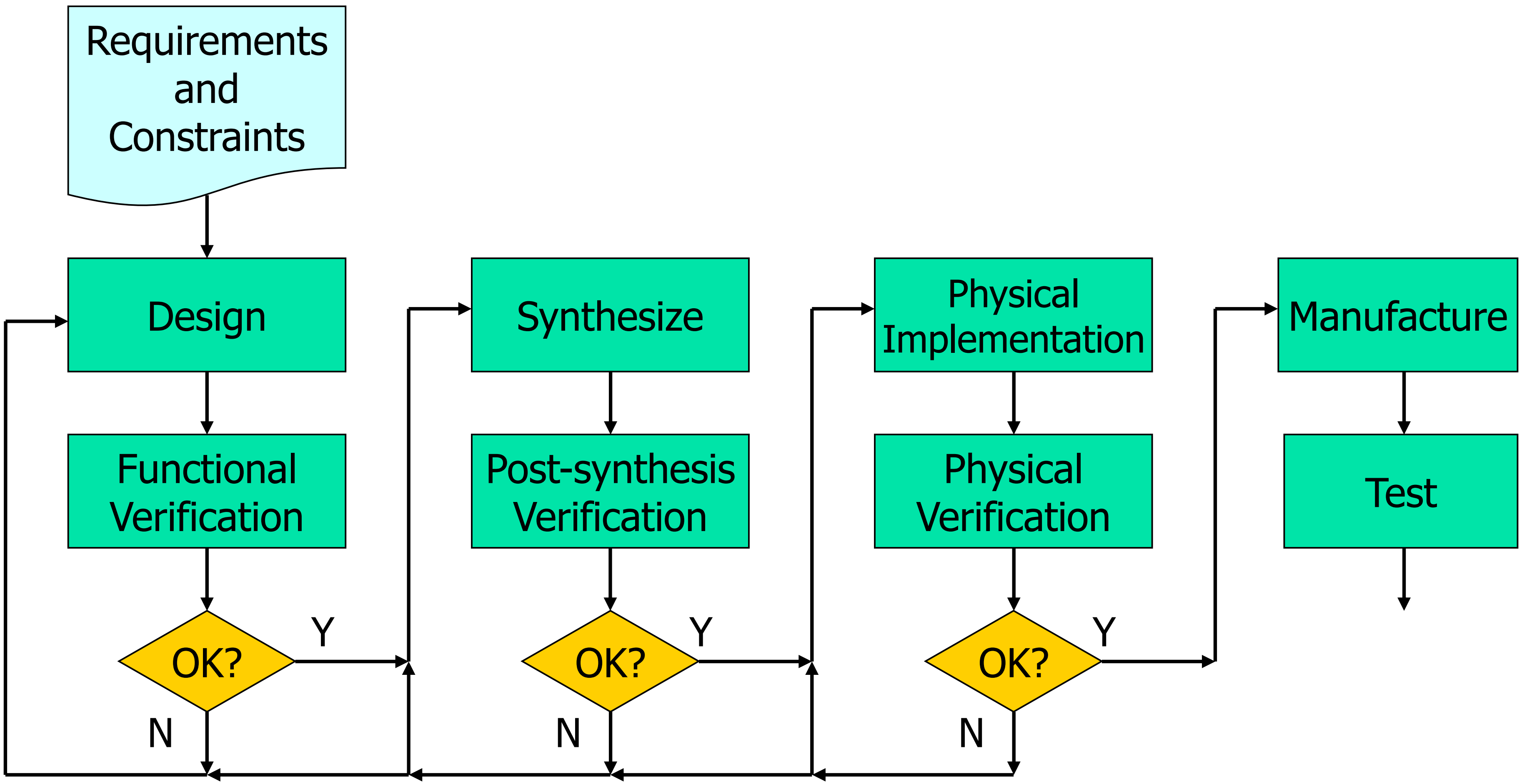
Hierarchical Design



What we learn by the end of semester

- Processor building blocks
  - Binary number representations
  - Types of Adders
  - Multipliers
  - Complex arithmetic hardware
  - Memories
- Communication algorithms and systems
- Design optimization targeted for FPGA
  - Verilog synthesis to a gate netlist
  - Delay estimation and reduction
  - Area estimation and reduction
  - Power estimation and reduction

A Simple Design Methodology



Hierarchical Design

- Circuits are too complex for us to design all the detail at once
- Design subsystems for simple functions
- Compose subsystems to form the system
  - Treating subcircuits as “black box” components
  - Verify independently, then verify the composition
- Top-down/bottom-up design

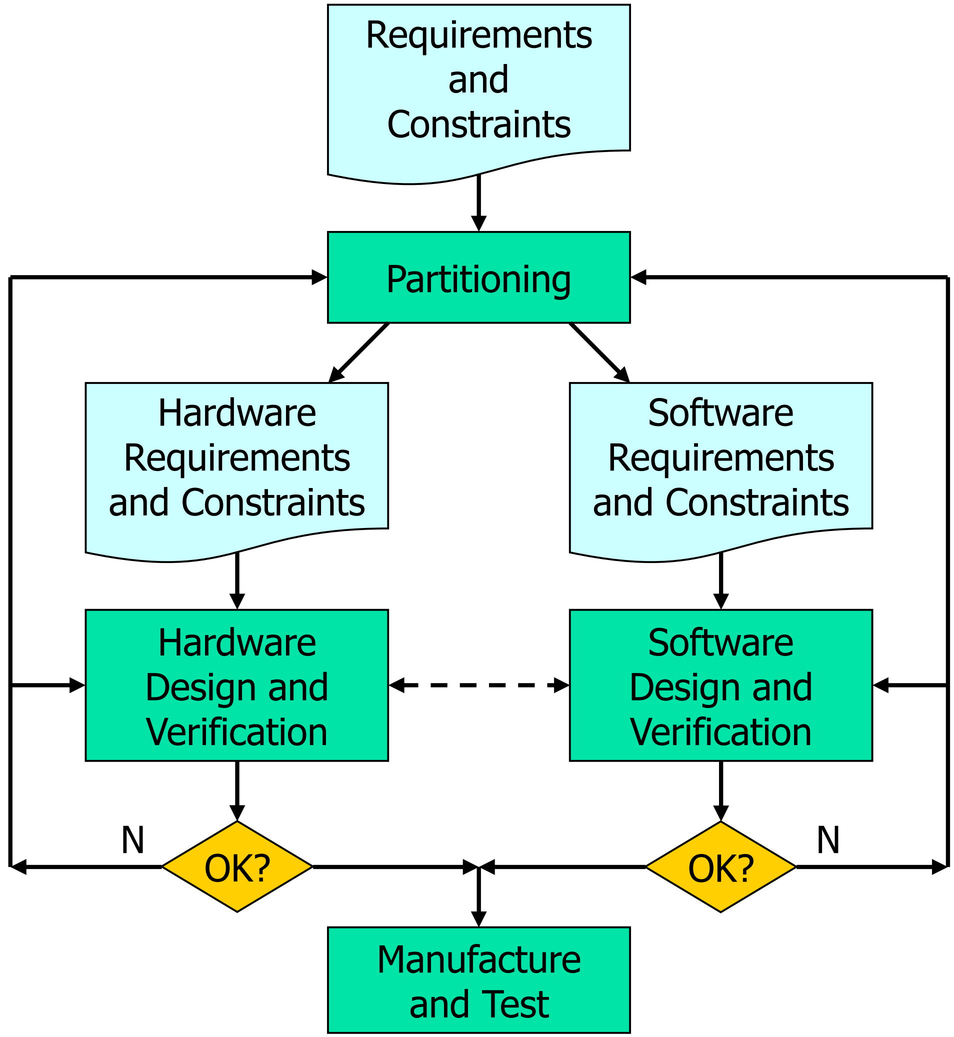
Synthesis

- We usually design using register-transfer-level (RTL) Verilog
  - Higher level of abstraction than gates
- Synthesis tool translates to a circuit of gates that performs the same function
- Specify to the tool
  - the target implementation fabric
  - constraints on timing, area, etc.
- Post-synthesis verification
  - synthesized circuit meets constraints

Physical Implementation

- Implementation fabrics
  - Application-specific ICs (ASICs)
  - Field-programmable gate arrays (FPGAs)
- Floor-planning: arranging the subsystems
- Placement: arranging the gates within subsystems
- Routing: joining the gates with wires
- Physical verification
  - physical circuit still meets constraints
  - use better estimates of delays

Codesign Methodology

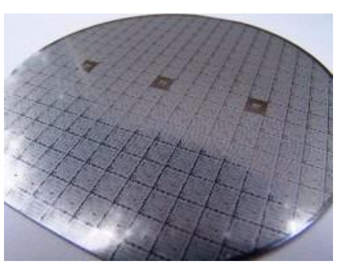


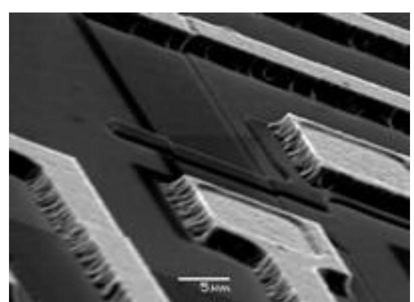
Summary

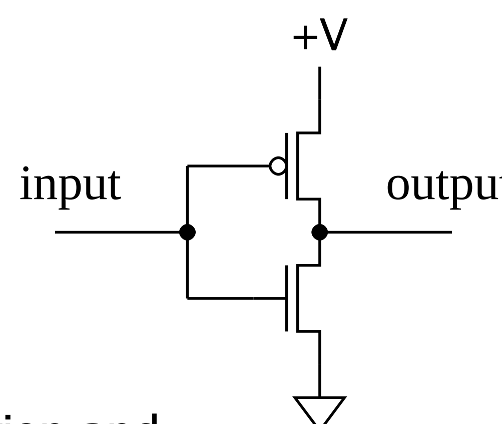
- Digital systems use discrete (binary) representations of information
- Basic components: gates and flipflops
- Combinational and sequential circuits
- Real-world constraints
  - logic levels, loads, timing, area, etc
- Verilog models: structural, behavioral
- Design methodology

Integrated Circuits (ICs)

- Circuits formed on surface of silicon wafer
  - Minimum feature size reduced in each technology generation
  - Currently 90nm, 65nm
  - Moore's Law: increasing transistor count
  - CMOS: complementary MOSFET circuits

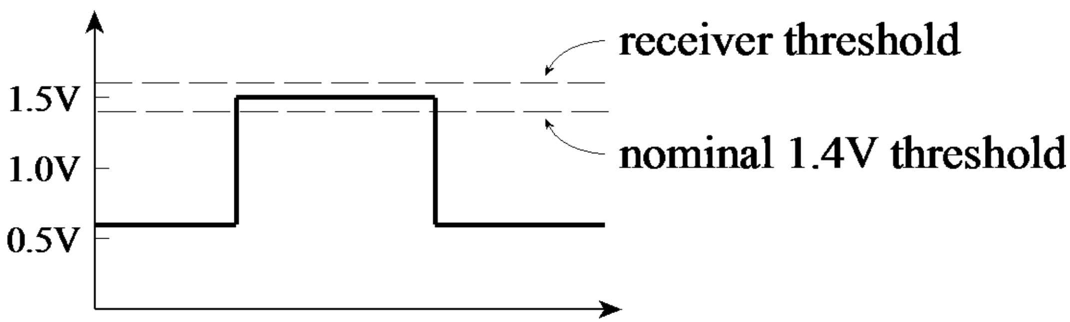


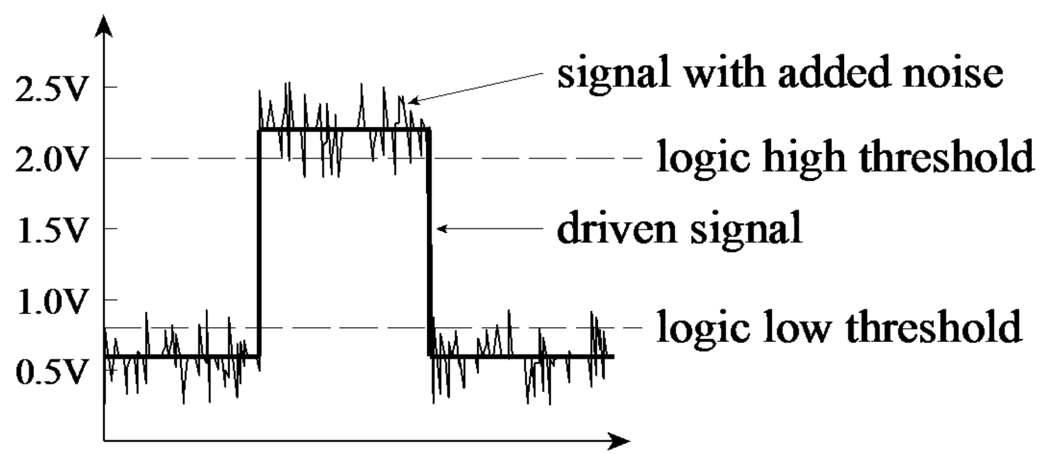




Logic Levels

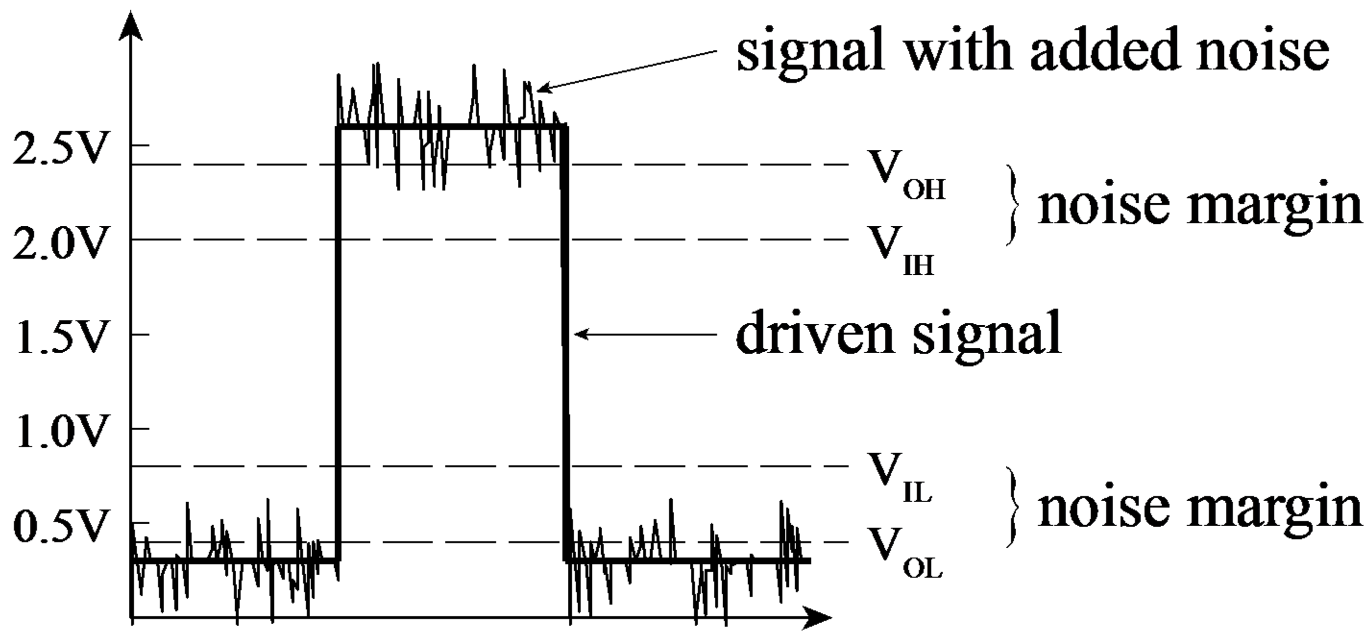
- Actual voltages for “low” and “high”
  - Example: 1.4V threshold for inputs





Logic Levels

- TTL logic levels with noise margins

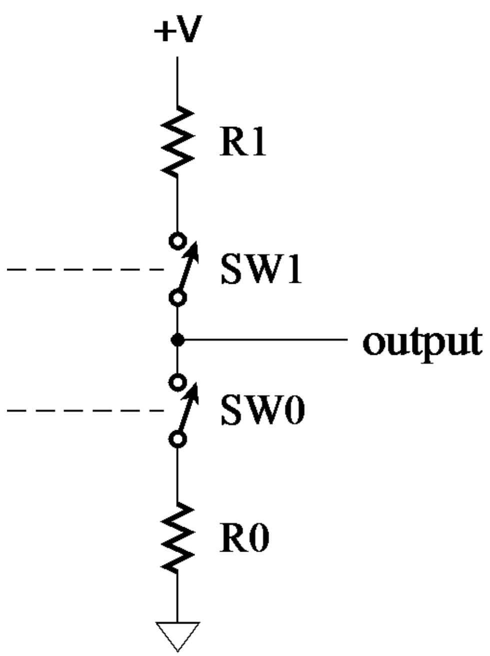


V_{OL}: output low voltage V_{IL}: input low voltage

V_{OH}: output high voltage V_{IH}: input high voltage

Static Load and Fanout

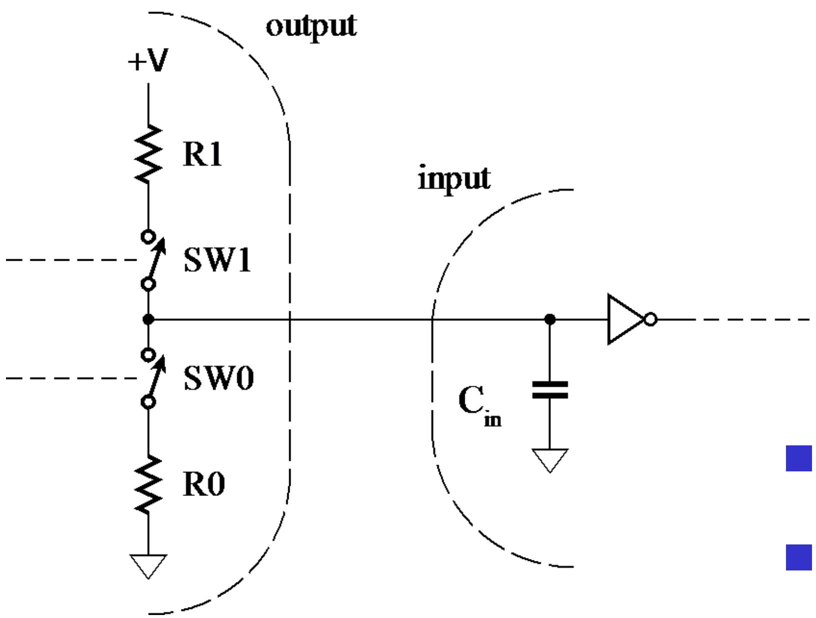
- Current flowing into or out of an output

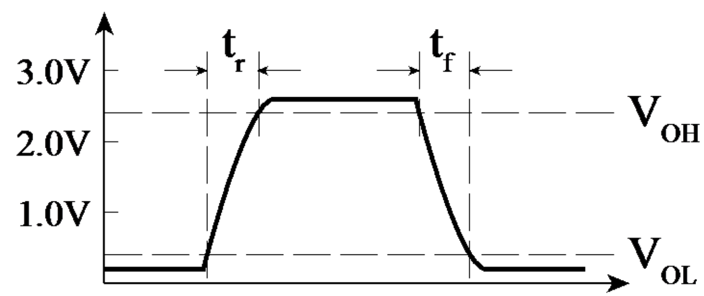


- High: SW1 closed, SW0 open
  - Voltage drop across R1
  - Too much current: $V_O < V_{OH}$
- Low: SW0 closed, SW1 open
  - Voltage drop across R0
  - Too much current: $V_O > V_{OL}$
- Fanout: number of inputs connected to an output
  - determines static load

Capacitive Load and Prop Delay

- Inputs and wires act as capacitors





- tr: rise time
- tf: fall time
- tpd: propagation delay
  - delay from input transition to output transition

Other Constraints

- Wire delay: delay for transition to traverse interconnecting wire
- Flipflop timing
  - delay from clk edge to Q output
  - D stable before and after clk edge
- Power
  - current through resistance => heat
  - must be dissipated, or circuit cooks!

Area and Packaging

- Circuits implemented on silicon chips
  - Larger circuit area => greater cost
- Chips in packages with connecting wires
  - More wires => greater cost
  - Package dissipates heat
- Packages interconnected on a printed circuit board (PCB)
  - Size, shape, cooling, etc, constrained by final product

Models

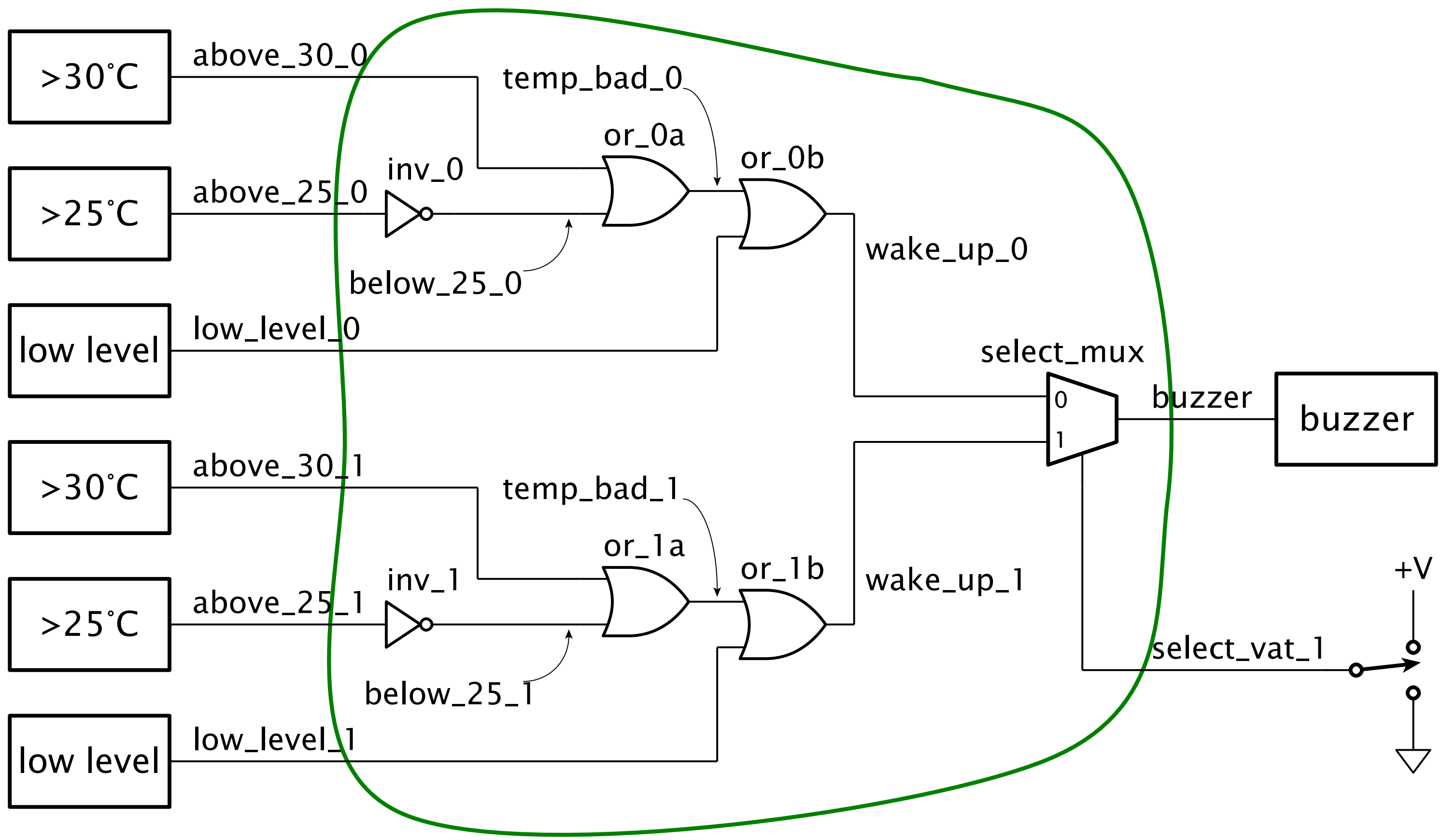
- Abstract representations of aspects of a system being designed
  - Allow us to analyze the system before building it
- Example: Ohm's Law
  - $V = I \times R$
  - Represents electrical aspects of a resistor
  - Expressed as a mathematical equation
  - Ignores thermal, mechanical, materials aspects

Verilog

- Hardware Description Language
  - A computer language for modeling behavior and structure of digital systems
- Electronic Design Automation (EDA) using Verilog
  - Design entry: alternative to schematics
  - Verification: simulation, proof of properties
  - Synthesis: automatic generation of circuits

Module Ports

- Describe input and outputs of a circuit



Structural Module Definition

```
module vat_buzzer_struct
  ( output buzzer,
    input above_25_0, above_30_0, low_level_0,
    input above_25_1, above_30_1, low_level_1,
    input select_vat_1 );

  wire below_25_0, temp_bad_0, wake_up_0;
  wire below_25_1, temp_bad_1, wake_up_1;

  // components for vat 0
  not inv_0 (below_25_0, above_25_0);
  or  or_0a (temp_bad_0, above_30_0, below_25_0);
  or  or_0b (wake_up_0, temp_bad_0, low_level_0);

  // components for vat 1
  not inv_1 (below_25_1, above_25_1);
  or  or_1a (temp_bad_1, above_30_1, below_25_1);
  or  or_1b (wake_up_1, temp_bad_1, low_level_1);

  mux2 select_mux (buzzer, select_vat_1, wake_up_0, wake_up_1);

endmodule
```

Behavioral Module Definition

```
module vat_buzzer_struct
  ( output buzzer,
    input above_25_0, above_30_0, low_level_0,
    input above_25_1, above_30_1, low_level_1,
    input select_vat_1 );

  assign buzzer =
    select_vat_1 ? low_level_1 | (above_30_1 | ~above_25_1)
                 : low_level_0 | (above_30_0 | ~above_25_0);

endmodule
```

Design Methodology

- Simple systems can be design by one person using *ad hoc* methods
- Real-world systems are design by teams
  - Require a systematic design methodology
- Specifies
  - Tasks to be undertaken
  - Information needed and produced
  - Relationships between tasks
    - dependencies, sequences
  - EDA tools used

Design using Abstraction

- Circuits contain millions of transistors
  - How can we manage this complexity?
- Abstraction
  - Focus on relevant aspects, ignoring other aspects
  - Don’t break assumptions that allow aspect to be ignored!
- Examples:
  - Transistors are on or off
  - Voltages are low or high

Embedded Systems

- Most real-world digital systems include embedded computers
  - Processor cores, memory, I/O
- Different functional requirements can be implemented
  - by the embedded software
  - by special-purpose attached circuits
- Trade-off among cost, performance, power, etc.