

Module 1: Introduction

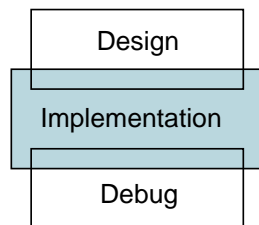
Digital Logic Design: ECEN 3233

Dr. Keith A. Teague

Fall 2004

Read Chapter 1, and Sections 3.1 & 3.2

The Process Of Design



Design

Initial concept: 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

Implementation

Assemble primitives into more complex building blocks
Composition via wiring
Choose among alternatives to improve the design

Debug

Faulty systems: design flaws, composition flaws, component flaws
Design to make debugging easier
Hypothesis formation and troubleshooting skills

The Art Of Design: Specifications and Constraints

1. Functional Specification/What the System Does

Ex: Traffic Light Controller

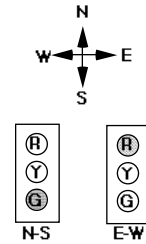
Lights point in the directions N, S, E, W

Illuminates the same lights N as S and E as W

Cycles thru the sequence GREEN-YELLOW-RED

N-S and E-W never GREEN or YELLOW at the same time

Stay GREEN for 45 seconds, yellow for 15, red for 60



2. Performance Constraints/Requirements to be Met

speed: compute changes in under 100 ms

power: consume less than 20 watts

area: implementation in less than 20 square cm

cost: less than \$20 in manufacturing costs

The Art of Design: "To Design Is To Represent"

1. English language specification

easy to write, but not precise and subject to ambiguity

2. Functional description

more precise specification

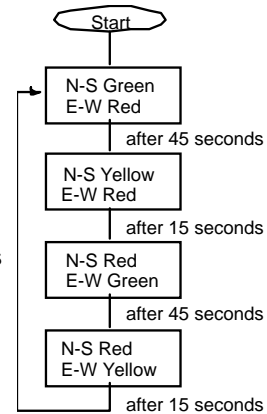
flow charts, program fragments

3. Structural description

complex components decomposed into compositions of less complex components

4. Physical description

the design in terms of most primitive building blocks, e. g., logic gates or transistors



The Process of Design: Debugging the System

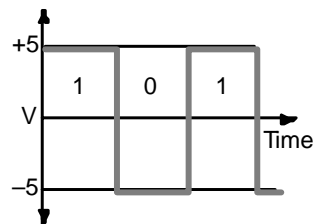
What Can Go Wrong

- **Design Flaws**
 - Implementation does not meet functional specification
 - Logic design is incorrect (wrong function implemented)
 - Misinterpretation or corner cases ignored
- **Implementation Flaws**
 - Individual modules function correctly but their compositions do not
 - Misunderstanding of interface and timing behavior
 - Wiring mistakes, Electrical mistakes
- **Component Flaws**
 - Logically correct and correctly wired
 - Not all hardware components are guaranteed to work!
E.g., burned out component

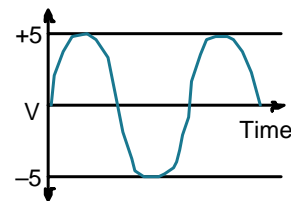
Digital Hardware Systems

Digital Systems

Digital vs. Analog Waveforms



Digital:
only assumes discrete values



Analog:
values vary over a broad range
continuously (a continuum)

The case shown is idealized – the waveform shown has only two values: +5 and -5 volts. In reality, logic-0 and logic-1 are represented by a range of voltages.

Digital Binary Systems - ideal

- **Two discrete values:**
yes, on, 5 volts, current flowing, magnetized North, "1"
no, off, 0 volts, no current flowing, magnetized South, "0"
- **Advantage of binary systems:**
rigorous mathematical foundation based on logic

IF the garage door is open
AND the car is running
THEN the car can be backed out of the garage

*both the door must
be open and the car
running before I can
back out*

IF N-S is green
AND E-W is red
AND 45 seconds has expired since the last light change
THEN we can advance to the next light configuration

The three preconditions must be true to imply the conclusion

Boolean Algebra and Logical Operators

Algebra: variables, values, operations

In Boolean algebra, the values are the symbols 0 and 1
If a logic statement is false, it has value 0
If a logic statement is true, it has value 1

Operations: AND, OR, NOT

| X | Y | X AND Y | X | Y | X OR Y | X | NOT X |
|---|---|---------|---|---|--------|---|-------|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | | |
| 1 | 1 | 1 | 1 | 1 | 1 | | |

Hardware Systems and Logical Operators

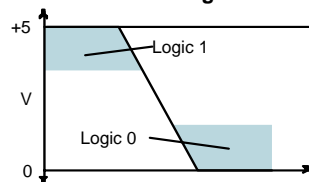
**IF the garage door is open
AND the car is running
THEN the car can be backed out of the garage**

| door open? | car running? | back out car? |
|------------|--------------|---------------|
| false/0 | false/0 | false/0 |
| false/0 | true/1 | false/0 |
| true/1 | false/0 | false/0 |
| true/1 | true/1 | TRUE/1 |

The Real World

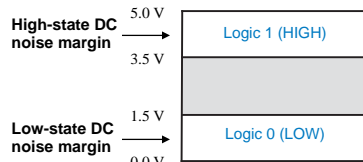
Physical electronic components are continuous, not discrete!

These are the building blocks of all digital components!



Transition from logic 1 to logic 0 does not take place instantaneously in real digital systems

Intermediate values may be visible for an instant



Switching threshold varies with voltage, temp, process, phase of the moon, etc. —need “noise margin”

Different electronic manufacturing technologies have different voltage characteristics

Boolean algebra useful for describing the steady state behavior of digital systems

Be aware of the dynamic, or time varying, behavior too!

Digital Circuit Technologies

Integrated circuit technology

choice of conducting, non-conducting, sometimes conducting ("semiconductor") materials

whether or not their interaction allows electrons to flow forms the basis for electrically controlled switches

Main technologies

MOS: Metal-Oxide-Silicon

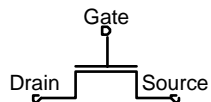
Bipolar

Transistor-Transistor Logic
Emitter Coupled Logic

MOS Technology

Transistor

basic electrical switch

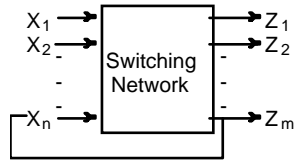


three terminal switch: gate, source, drain

- ❖ voltage between gate and source exceeds threshold
- ❖ switch is conducting or "closed"
- ❖ electrons flow between source and drain

- ❖ when voltage is removed -
- ❖ the switch is "open" or non-conducting
- ❖ connection between source and drain is broken

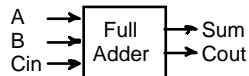
Combinational vs. Sequential Logic



Network implemented from switching elements or logic gates. The presence of feedback distinguishes between *sequential* and *combinational* networks.

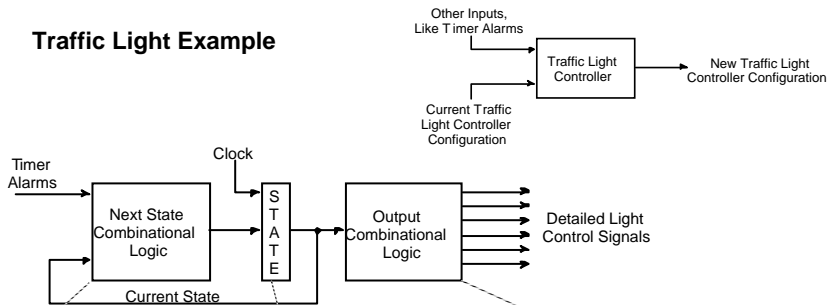
Combinational logic

- no feedback among inputs and outputs
- outputs are a pure function of the inputs
- e.g., full adder circuit:
(A, B, Carry In) mapped into (Sum, Carry Out)



Combinational vs Sequential Logic

Traffic Light Example



Next State Logic

Maps current state and alarm events into the next state

Current State

Storage elements replaced by next state when the clock signal arrives

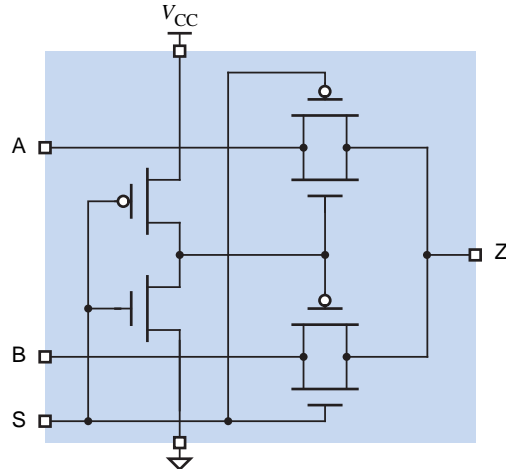
Output Logic

Current state mapped into control signals to change the lights and to start the event timers

IF controller in state N-S green, E-W red
AND the 45 second timer alarm is asserted
THEN the next state becomes N-S yellow,
E-W red when the clock signal is next asserted

- Transistor-level circuit diagrams
- Low level
 - Detailed
- Electronic, not logic

This circuit is called a 2:1 multiplexer



Truth Tables

tabulate all possible input combinations and their associated output values

Example: half adder
adds two binary digits to form Sum and Carry

| A | B | Sum | Carry |
|---|---|-----|-------|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

NOTE: 1 plus 1 is 0 with a carry of 1 in binary

Example: full adder
adds two binary digits and Carry in to form Sum and Carry Out

| A | B | Cin | Sum | Cout |
|---|---|-----|-----|------|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

Representations of a Digital Design

Boolean Algebra

values: 0, 1
variables: A, B, C, . . . , X, Y, Z
operations: NOT, AND, OR, . . .

NOT X is written as \bar{X}
X AND Y is written as X & Y, or sometimes X Y
X OR Y is written as X + Y

Deriving Boolean equations from truth tables:

| A | B | Sum | Carry |
|---|---|-----|-------|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

$Sum = \bar{A} B + A \bar{B}$

$Carry = A B$

OR'd together *product* terms for each truth table row where the function is 1

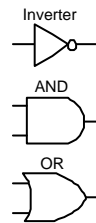
if input variable is 0, it appears in complemented form;
if 1, it appears uncomplemented

Representations of a Digital Design

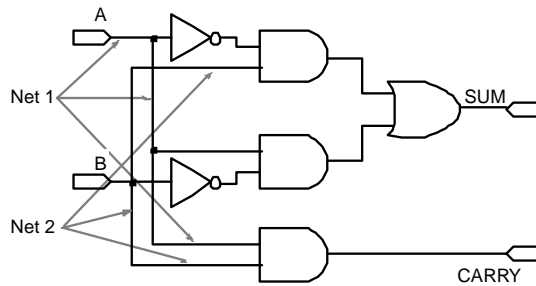
Logic Diagrams / Gates / Schematics

Gates: most widely used primitive building block in digital system design

Standard Logic Gate Representation



Half Adder Schematic (logic diagram)



Net: electrically connected collection of wires

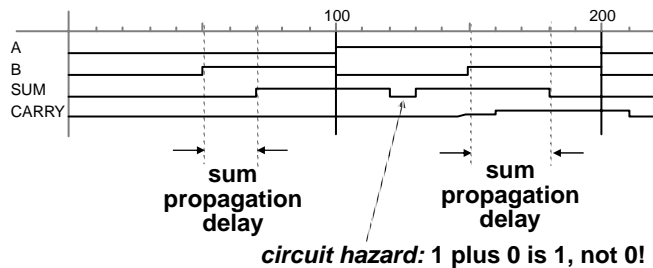
Netlist: tabulation of gate inputs & outputs and the nets they are connected to

Representations of a Digital Design

Waveforms

dynamic behavior of a circuit
real circuits have non-zero delays

Timing Diagram of the Half Adder



Output changes are delayed from input changes

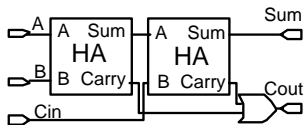
The propagation delay is sensitive to paths in the circuit

Outputs may temporarily change from the correct value to the wrong value back again to the correct value: this is called a *glitch* or *hazard*

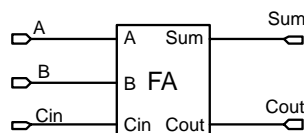
Representations of a Digital Design

Function Blocks

- structural organization of the design
- black boxes with input and output connections
- corresponds to well defined functions
- concentrates on how the components are composed by wiring



Full Adder realized in terms of composition of half adder blocks



Block diagram representation of the Full Adder

- Various hardware description languages

- ABEL
- VHDL
- Verilog

- We'll avoid HDL and work mainly from logic diagrams; however, the Xilinx software generates VHDL descriptions internally!

```
module chap1mux  
title 'Two-input multiplexer example'  
CHAP1MUX device 'P16V8'
```

```
A, B, S    pin 1, 2, 3;  
Z          pin 13 istype 'com';
```

```
equations
```

```
WHEN S == 0 THEN Z = A; ELSE Z = B;
```

```
end chap1mux
```

```
library IEEE;  
use IEEE.std_logic_1164.all;
```

```
entity Vchap1mux is  
port ( A, B, S: in  STD_LOGIC;  
      Z:      out STD_LOGIC );  
end Vchap1mux;
```

```
architecture Vchap1mux_arch of Vchap1mux is  
begin  
  Z <= A when S = '0' else B;  
end Vchap1mux_arch;
```

Simulation

program which dynamically executes an abstract design description

obtain verification of functional correctness and some timing information before the design is physically constructed

easier to probe and debug a simulation than an implemented design

simulation cannot guarantee that a design will work

only as good as the test cases attempted

does not check electrical errors

abstracts away some of the realities of a real system

Logic Simulation (Logicworks, Xilinx Tools)

design described in terms of logic gates

values are 0, 1 (plus others to be introduced)

good for truth table verification

Timing Simulation (Xilinx Tools)

waveform inputs and outputs

model of gate delays

are the waveform shapes what was expected?

identification of performance bottlenecks

Chapter Review

We have introduced:

- *The process of design:*
functional decomposition and design by assembly

- *The kinds of systems we will be designing:*
combinational and sequential logic
binary digital systems
implemented in MOS and bipolar technology

- *The many levels of design representation:*
from gates to behavioral descriptions

- *The changing technological landscape:*
rapid electronic system implementation
facilitated by computer-aided design tools
(in particular, synthesis and simulation tools)