Introduction to Digital Logic

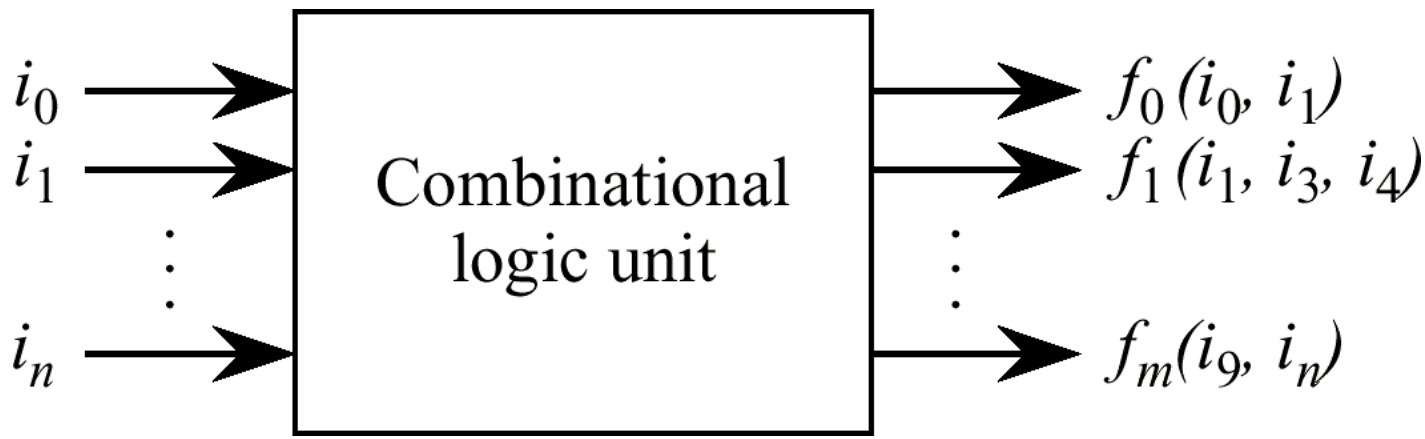
Computer Organization at the lowest level

Objectives

- After this lecture, you should be able to…….
  - **State the 3 possible arrangements of digital logic**
  - **Formulate simple electronic logic gate function**
  - **Define what is meant by a *truth table***
  - **Write truth tables for 2-input AND, OR, NOT forms**
  - **Interpret Boolean expressions in software form**
  - **Synthesize composite combinational functions**
  - **Develop truth tables from gate level diagrams**
  - **Recognize the importance of power dissipation and propagation delay in logic circuits**

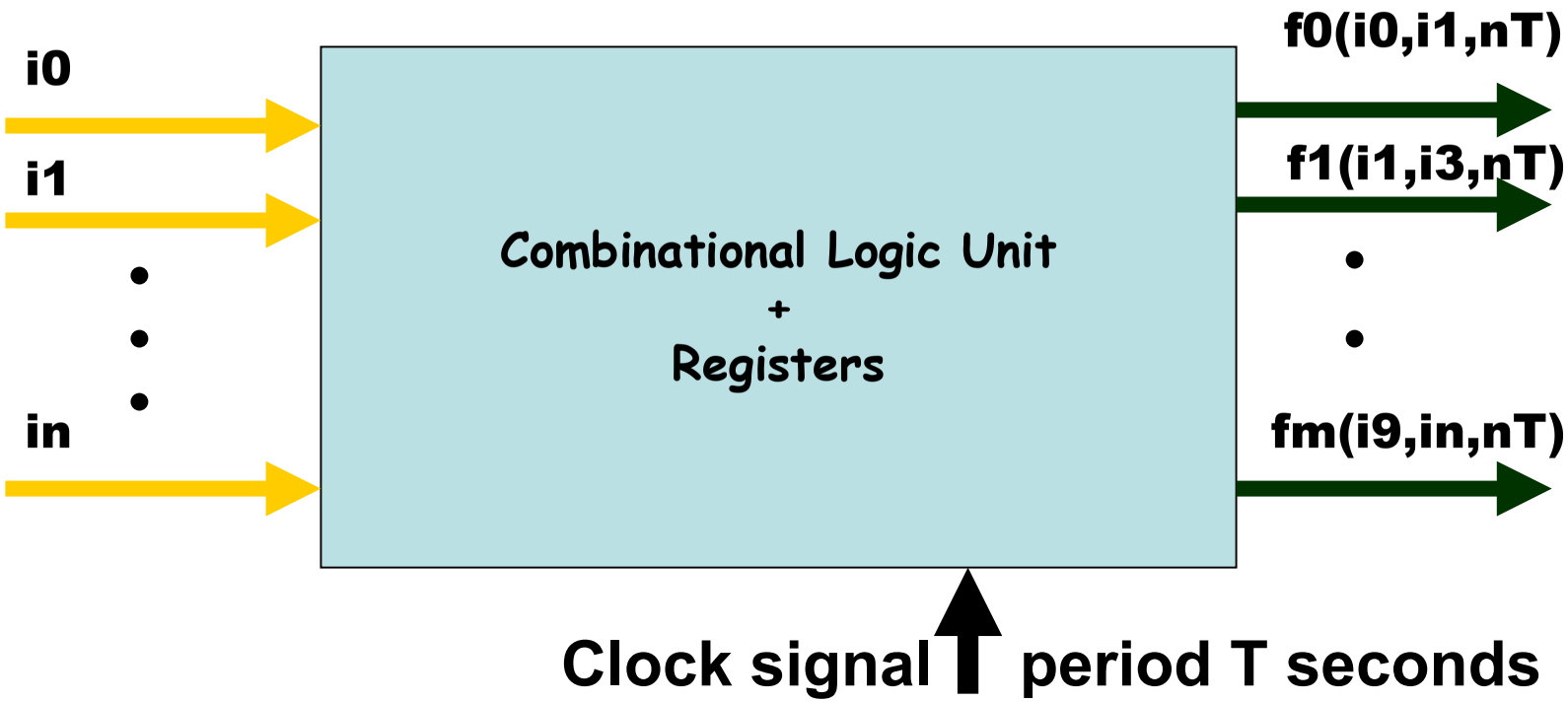
The Combinational Logic Unit

- Translates a set of inputs into a set of outputs according to one or more mapping functions.
- Inputs and outputs for a CLU normally have two distinct (binary) values: high and low, 1 and 0, 0 and 1, or 5 v. and 0 v. for example.
- The outputs of a CLU are strictly functions of the inputs, and the outputs are updated immediately after the inputs change. A set of inputs i0 – in are presented to the CLU, which produces a set of outputs according to mapping functions f0 – fm



Sequential Logic

- **Outputs now depend not only on the current state of the inputs but also the PAST HISTORY of the inputs. This kind of logic incorporates MEMORY or REGISTER functions in order to work differently from combinational logic.**



Another Definition…..

- **Combinational logic: a system in which logical decisions are made based only on COMBINATIONS of the CURRENT INPUTS e.g. an adder.**
- **Sequential logic: a system in which decisions are made based on COMBINATIONS of the CURRENT INPUTS as well as the PAST HISTORY of inputs. e.g. a memory unit.**
- **Finite state machine: in this model, logic maintains an internal state; Its outputs are functions of both CURRENT INPUTS and its INTERNAL STATE e.g. of a physical implementation of a FSM is a vending machine controller.**

Making Logic Gate Circuits

- **We will review material located at the following URL:**
  http://www.spsu.edu/cs/faculty/bbrown/web_lectures/index.html
- **Bob Brown (SCSE Southern Polytechnic State University) has prepared some good interactive notes**
- **We will review the first two sections of the "Gates Circuits and Digital Logic" Chapter:**
  - **Electric Circuits**
  - **Transistors and Digital Logic Gates**
- **You are encouraged to re-read these later and to refer to the relevant sections of the course textbooks**

Boolean Operators in Programs

- We will now learn to analyze some written Boolean operator statements from the programmer's viewpoint
  - OR function
  - AND function
  - Boolean expressions

Boolean OR Operator

- Evaluate this logical expression
  - If ((x > 0) || (x < 5)) y = 10;
  - x is assigned the value of 7
  - If (TRUE || FALSE) y =10;
  - If TRUE y = 10;
  - y=10;

Boolean AND Operator

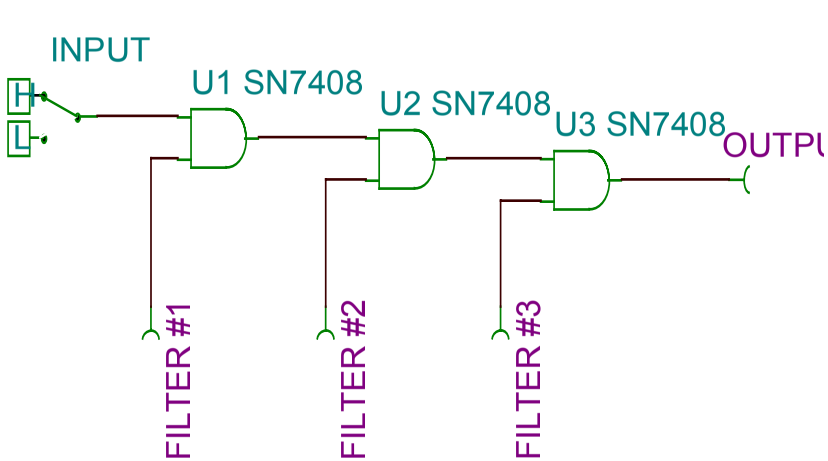
- Let's look at how this conditional statement works
  - If ((x > 0) && (x < 5)) z = 20;
  - Again we assign a value of 7 to variable x
  - If (TRUE && FALSE) z =20;
  - If FALSE z = 20;
  - This is a null statement
  - No value is assigned to z. Nothing happens.

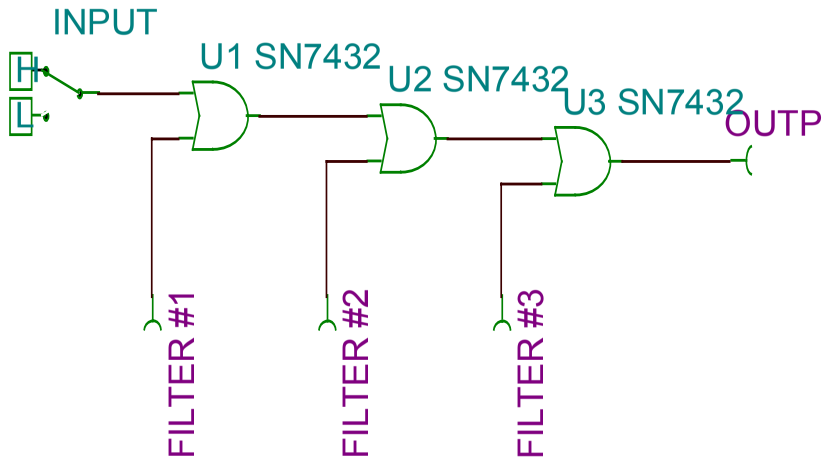
Boolean Expressions – An Example

- Work on the Boolean expression in NeveIn on p 32

Closed and Open Datapaths

- Let's look at two chains of logic and see how they process inputs (data)

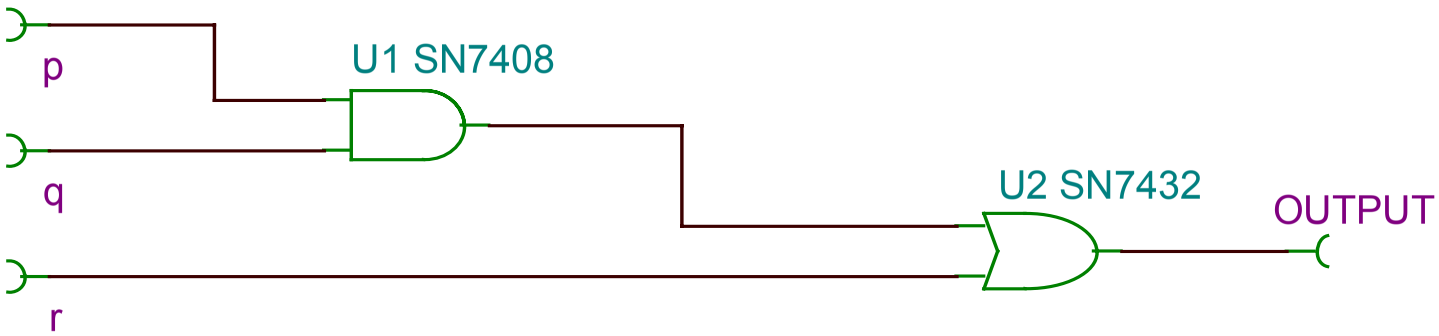




How is each activated to pass changing inputs faithfully to the output?

Boolean Networks

- OUTPUT = ((p AND q) OR r)
  - An example of a composite function which can be synthesized with simple logic forms { AND, OR }



Synthesizing XOR function

- We will use 2-input NOT, AND and OR gates to synthesize this function
- We look for ways of asserting a TRUE condition (logical ‘1’) output in the truth table
- We then logically combine these conditions

XOR Synthesis continued

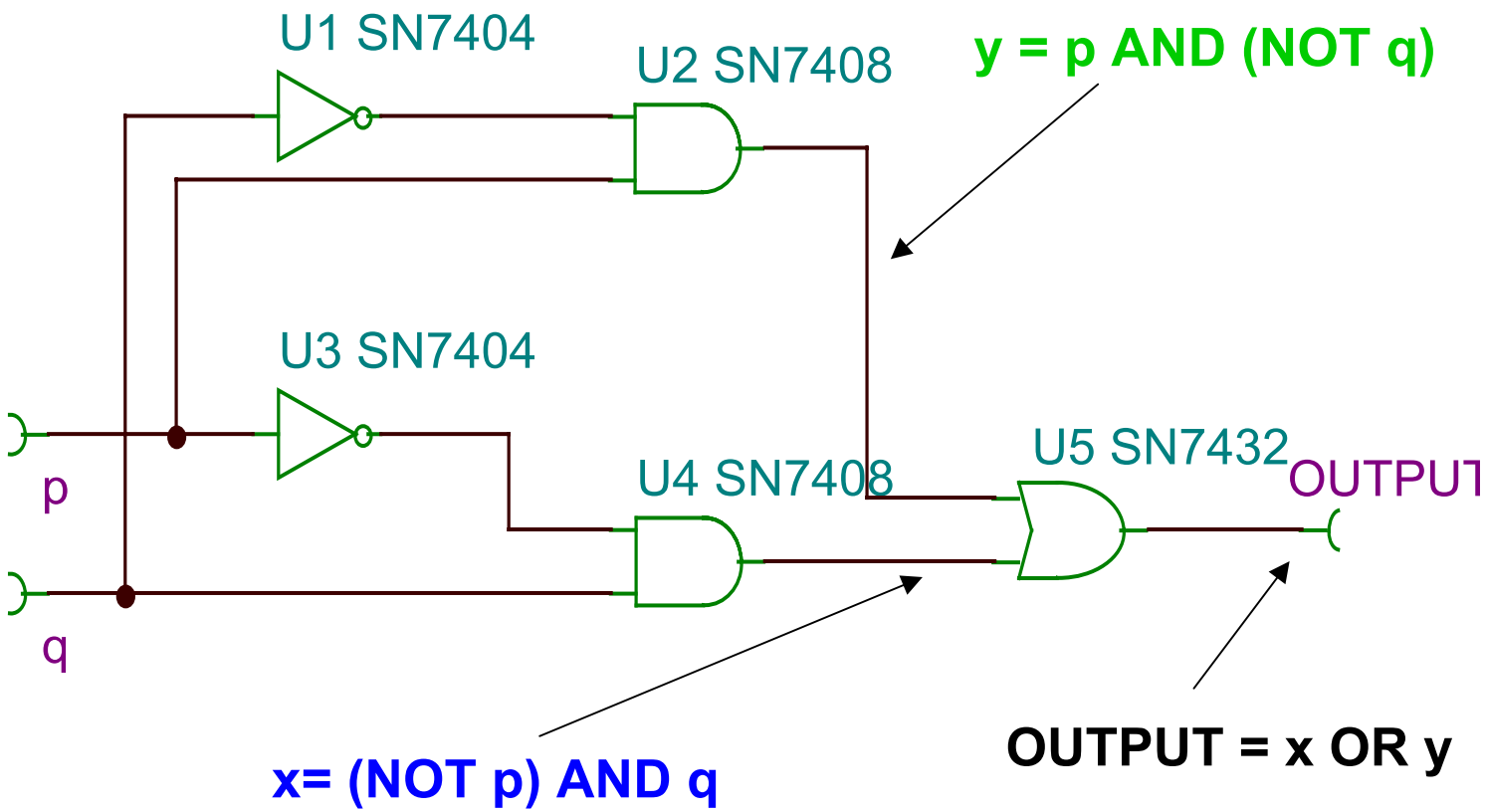
p	q	(p XOR q)
0	0	0
0	1	1
1	0	1
1	1	0

x= (NOT p) AND q

y = p AND (NOT q)

OUTPUT = x OR y

Completed XOR Synthesis



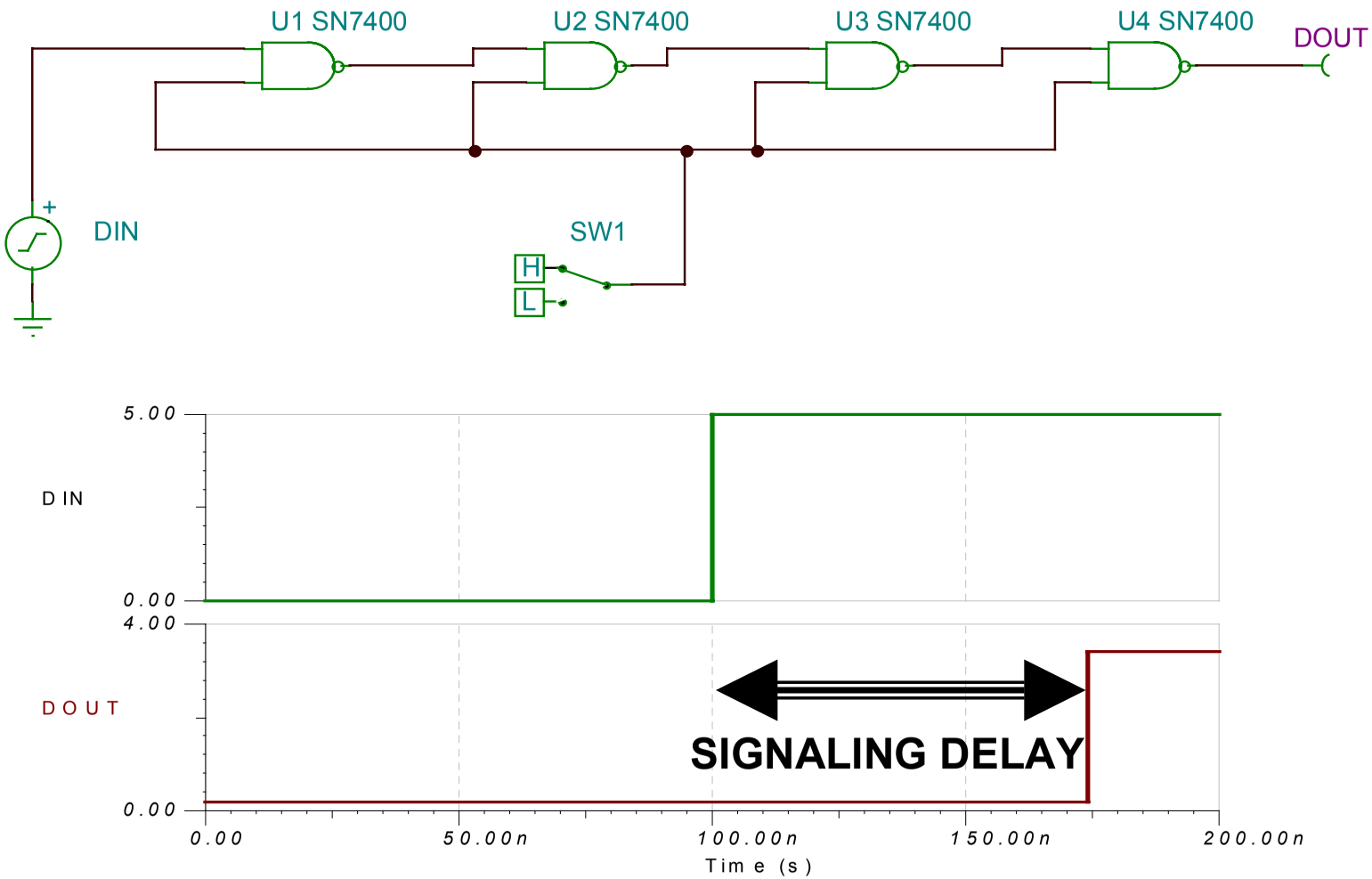
Propagation Delay

- This determines how quickly logic can change state
- Sets the CLOCK SPEED because combination logic and sequential logic must work correctly TOGETHER in complex logic blocks

Comparing Clock Frequency and Cycle Time

Frequency		Cycle Length	
1 Hz	1 cps	1 s	1 sec
1 kHz	10^3 cps	1 ms	1/1,000 sec
1 MHz	10^6 cps	1 us	1/1000,000 sec
1 GHz	10^9 cps	1 ns	1/1000,000,000 sec

illustrating Propagation Delay



Power Dissipation

- When IC gates are switching, they transfer charge from
  - Either the DC power supply to the "load" i.e. gates attached to their output terminal….
  - ….or from the load to the ground return
- Any motion of charge in a circuit is a current (measured in units of Amperes)
- Any current I in a circuit under a potential difference V (volts) is power dissipated - P = I x V Watts. This produces heat.
- Heat must be moved off the chip to keep the IC's temperature within safe operating limits.

Let's Check off Objectives

1. **Reviewed 3 digital logic classifications**
2. **Modeled simple gate functions**
3. **Defined the term truth table**
4. **Wrote truth tables for NOT, OR, AND**
5. **Mapped logic gate circuit to EXOR**
6. **Reviewed cascaded OR, AND logic**
7. **Identified the TWO P's - critical issues that impact computers at the gate level**
    1. **Propagation Delay**
    2. **Power Dissipation**