Digital Electronics

Introduction:

In electronics we can classify the building blocks of a circuit or system as being either analog or digital in nature. If we focus on voltage as the circuit parameter of interest then we can define analog and digital in the following fashion:

Analog: The voltage can take on a range of voltages, e.g. any value between 0.1 and 2 Volts.

Digital: The voltage can have only two values, e.g. 0 or 5 Volts, or we can say the voltage is either on or off and not even give specific values for the voltage when the circuit is on or off.

Digital circuits are useful when we don't need a continuous range of voltage or current. Some examples are: representing numbers, binary logic, counting circuits.

Example: Representing numbers with components that are either ON (1) or OFF (0).

Represent base 10 numbers using the binary system.

 $2I_{10} = 10I_2 = 1 \times 2^1 + 0 \times 2^0$ $10I_{10} = 1010I_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$

Digital circuits use standard voltages (or currents) to denote ON (high, 1) or OFF (low, 0). These standards are called "Logic Families" and there are several families. Two of the most popular families are:

TTL (Transistor-Transistor-Logic): ON = 5 Volts, OFF = 0 Volts

ECL (Emitter-Coupled-Logic): ON = -1 Volt, OFF = -1.6 Volts

For practical reasons both ON and OFF are given by a range of voltages or currents. Also ON for an input to a circuit might be a slightly different voltage than ON for an output to a circuit.

A description of several logic families is given in the table below:

Standard TTL (7400)	Delay (ns) 10	Max. FF Rate (MHz) 35	Power/Gate (mW) 15	High (V) 3.5	Low (V) 0.2
Low-power Schottky (74LS00)	9.0	33	2	3.5	0.2
Fast TTL (74F00)	3.5	125	5.5	2.7	0.5
CMOS (74C00)	25 @ 10 V 50 @ 25 V	10 @ 10V 3.5 @ 5 V	0.01	5-15	0
High-speed CMOS (74HC00)	8.0	40	0.01	2-6	0.1
ECL	2	250	25	-0.9	-1.8
100k ECL	0.75	500	40	-1.0	-1.7

Advantages of Digital:

a) only deal with two voltage levels (either ON or OFF)

b) voltages (or currents) are standardized

c) do not deal with individual transistors.....

Disadvantages of Digital:

a) too many "black" boxes

b) need good power supplies, clocks etc. for circuits to work properly

Logic Gates:

We want to make decisions based on digital information. For now consider the basic building blocks with one or two inputs and one output.

The basic logic units (gates) are: AND, OR, NOT. These functions are defined by their truth tables. A and B stand for the inputs, Y is the output. A 0 is a low input or output, a 1 is a high input or output.



Boolean Algebra or the Algebra of 1's and 0's

Circuits consisting of logic gates are described by Boolean algebra. A little use of this algebra can go along way in helping simplify circuit design, e.g. minimize the number of components.

The following theorems can be proved using a truth table and the definition of OR, AND, and NOT. See Simpson page 540 for more theorems.

1) $\overline{A} + \overline{A} = \overline{A}, A + 1 = 1, A + 0 = A$ 2) AA = A3) AB = BA4) ABC = (AB)C = A(BC)5) A(B + C) = AB + AC6) $\overline{1} = 0, \overline{0} = 1$ 7) $A + \overline{A} = 1, A\overline{A} = 0, A \cdot 1 = A$ 8) $\overline{\overline{A}} = A$ 9) $\overline{A + B} = \overline{A} \cdot \overline{B}$ 10) $\overline{AB} = \overline{A} + \overline{B}$

For the sake of clarity I have left the (AND symbol) out of some of the above theorems. Theorems 9) and 10) are also known as DeMorgan's Theorem.

Example using Boolean algebra:

Prove: X + YZ = (X + Y)(X + Z) (X + Y)(X + Z) = XX + XZ + YX + YZ by 5) = X + X(Z + Y) + YZ by 2) and 5) = X(1 + Z + Y) + YZ by 5) = X + YZ by 1)

Note: We could also have proven the above using a truth table. In this case there are 8 (2³) possible combinations of X, Y, Z. For a large number of inputs using a truth table becomes unwieldy. For example, if we had 10 inputs then there would be $2^{10} = 1024$ possible combinations to consider!

Example: The Exclusive $OR = XOR = A \oplus B$.

Output is high if inputs are different.



How do we make an exclusive OR with AND, OR, and NOT gates? Brute Force method:



Can we simplify this circuit? simplify = use less parts Use logical theorems:

$\mathbf{A} \oplus \mathbf{B} = \mathbf{A}\overline{\mathbf{B}} + \overline{\mathbf{A}}\mathbf{B}$		
$= \overline{AA} + \overline{AB} + \overline{AB} + \overline{BB}$	7) and 1)	
$= A(\overline{A} + \overline{B}) + B(\overline{A} + \overline{B})$	5)	
= $A(\overline{AB}) + B(\overline{AB})$	10)	
$= (A + B)(\overline{AB})$	5)	
A+B		



This circuit uses only 3 parts (OR, NAND, AND), but each of them is different! Usually there are many ways to synthesize the same function (circuit). Must decide if you want to minimize:

Must decide if you want to minimize: number of components types of components number of connections power consumption

For example we can make an XOR using only 4 NAND gates:



Final example: Suppose you have a light controlled by 3 switches. You want the light to be on if any one of the 3 switches is on or if all 3 switches are on.



Flip-Flops:

Basic counting unit in computer: counters shift registers memory Circuit whose output depends on the history of its inputs. Can make a flip-flop with just 2 transistors (or 2 vacuum tubes 1919!). Lots of different types of flip-flops (e.g. RS, JK, T, D).

Example: RS flip-flop or Reset-Set flip flop

Flip-flops, like logic gates are defined by their truth table. Flip-flops are controlled by an external clock pulse, all inputs and outputs are logic levels (e.g. TTL, ECL). Can make an RSFF out of NOR gates:





 Q_n is the present state of the FF.

 Q_{n+1} will be the output after the clock enables the FF to look at its inputs (R and S). Many FF change state $(Q_n \rightarrow Q_{n+1})$ on the trailing edge of the clock.

Note: The state with R = S = 1 is undefined. The output is not predictable!

Example: JK flip-flop

JKFF is like the RSFF except that both inputs (J and K) can be high (1).



Most JKFF's have a connection for forcing Q = 0 (clear) or forcing Q = 1 (preset)

Example: Counter made from two JK flip-flops.

This circuit counts from 0 1 2 3 0 1 2 3 0.....

 Q_1 is the lowest order bit, Q_2 is the higher order bit

The output is a binary number = $Q_2 Q_1$.

The o's on the clock means that the output transition occurs on the trailing edge of the clock pulse.



Input high = 1Output1 0 1 0 1 Q_1 low = 0Output2 0 0 1 1 Q,

Output of circuit is most conveniently displayed using a timing diagram:

Time

Flip-Flops are a class of circuits called "multivibrators"

Multivibrators are circuits with one or more stable states.

There are 3 common varieties of multivibrators:

1) Monostable multivibrators (one shot) have one stable state. If the circuit is forced out of its stable state (by e.g. an input pulse) it eventually returns back to the stable state by itself.

2) Bistable multivibrators have two stable states. Transitions between the states occur only by an external action (e.g. voltage pulse for flip-flops). Transition voltages can be different for the two states (e.g. Schmitt trigger).

3) Astable multivibrators are two state devices which switch on their own accord. Commonly used as oscillators.

Example: Bistable Multivibrator. This circuit has two stable states.

When either transistor conducts there is 4 mA flowing (by design) in the collector (I_{C1} or I_{C2})

transistor 1 off, transistor 2 on State 1: $V_{C1} \approx 11 \text{ V}, V_{B1} \approx 2 \text{ V}, V_{E1} \approx 4 \text{ V}$ $V_{C2} \approx 4 \text{ V}, V_{B2} \approx 5.5 \text{ V}, V_{E2} \approx 4 \text{ V}$ note: T_2 is saturated ($V_{CE} \approx 0$)

Transition from state 1 to state 2:

Input pulse forces T_1 to conduct, T_1 conducting means that V_{C1} drops.

 V_{C1} causes V_{B2} to drop to the point where T_2 is not conducting.

transistor 1 on, transistor 2 off State 2: $V_{C1} \approx 4 \text{ V}, V_{B1} \approx 5.5 \text{ V}, V_{E1} \approx 4 \text{ V}$ note: T₁ is saturated ($V_{CE} \approx 0$) $V_{C2} \approx 11 \text{ V}, V_{B2} \approx 2 \text{ V}, \widetilde{V}_{E2} \approx 4 \text{ V}$



