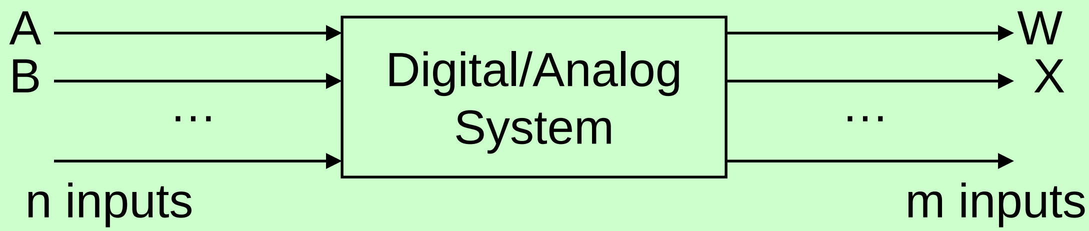
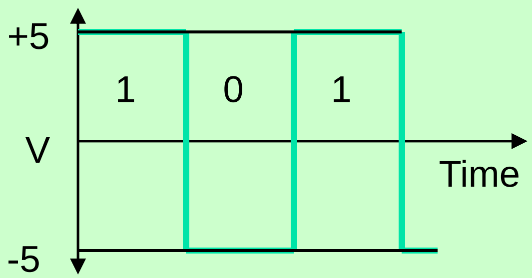
Digital Systems

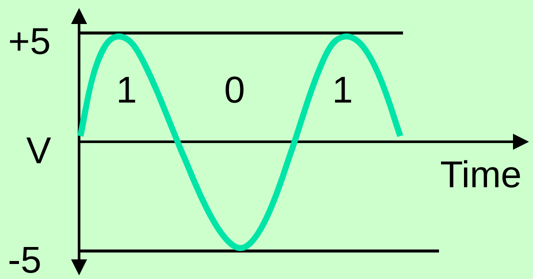
- Digital systems vs. Analog systems



  - Digital systems: input and output signals are represented by discrete values
  - Analog systems: input and output signals take on a continuous range of values



digital waveform



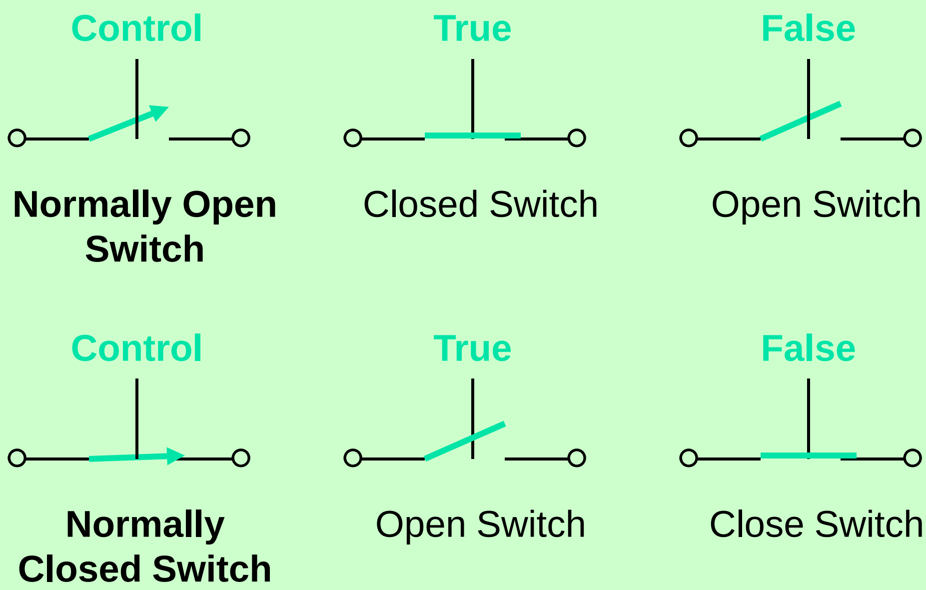
analog waveform

Binary Digital Systems

- The simplest form of digital systems
- Inputs and outputs are coded in binary digits (bits), “1” or “0”
- Binary values “1” and “0” are easy to encode with physical quantities:
  - Voltages: “1”- 5V, “0”- 0V
  - Magnetic polarizations: “1”- North, “0”- South

Representations of Digital Design

- Switches



Representations of Digital Design

- Truth tables
  - e.g.: Half adder

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

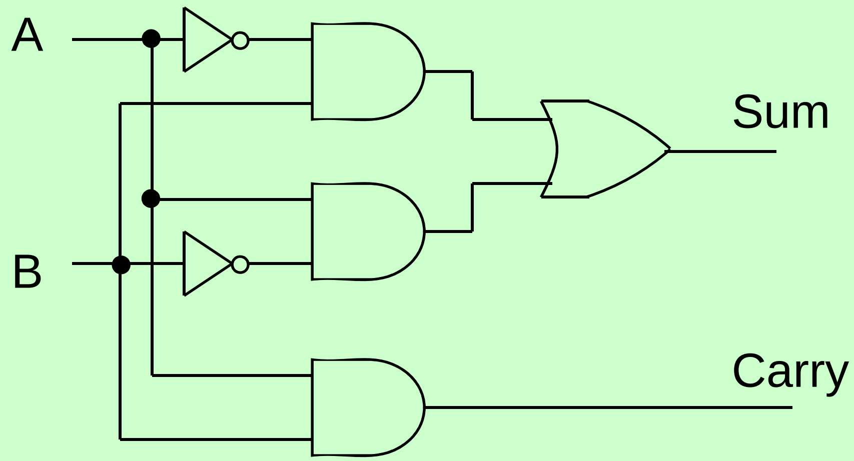
- Boolean Algebra
  - e.g.: Half adder

Sum = (AB') + (A'B)

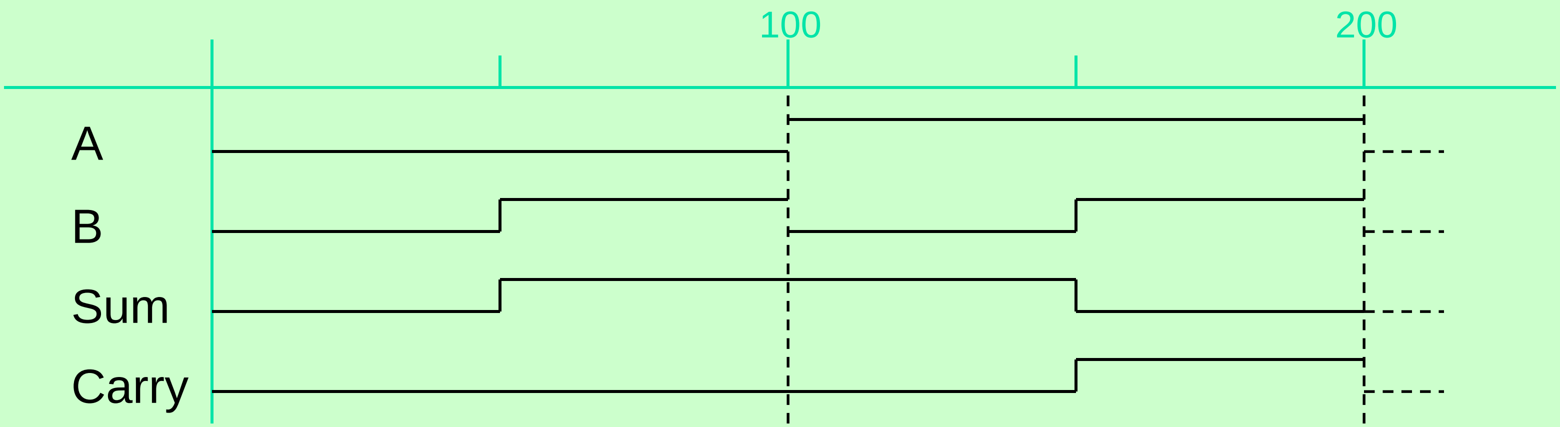
Carry = AB

Representations of Digital Design

- Gates
  - e.g.: Half adder

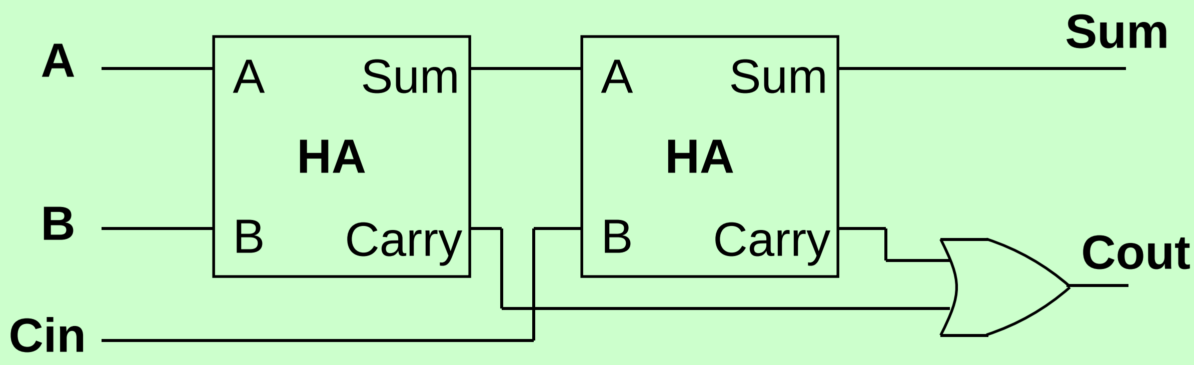


- Waveforms



Representations of Digital Design

- Blocks



Block Diagram of Full Adder

- Behaviors
  - Hardware description languages: VHDL, Verilog

Digital Subsystems

- Combinational logic
  - Output depends only on the present value of the input
  - Chapter 3
- Sequential logic
  - Output is determined by the current input and the previous state of the circuit
  - Input + memory → output
  - Chapter 4

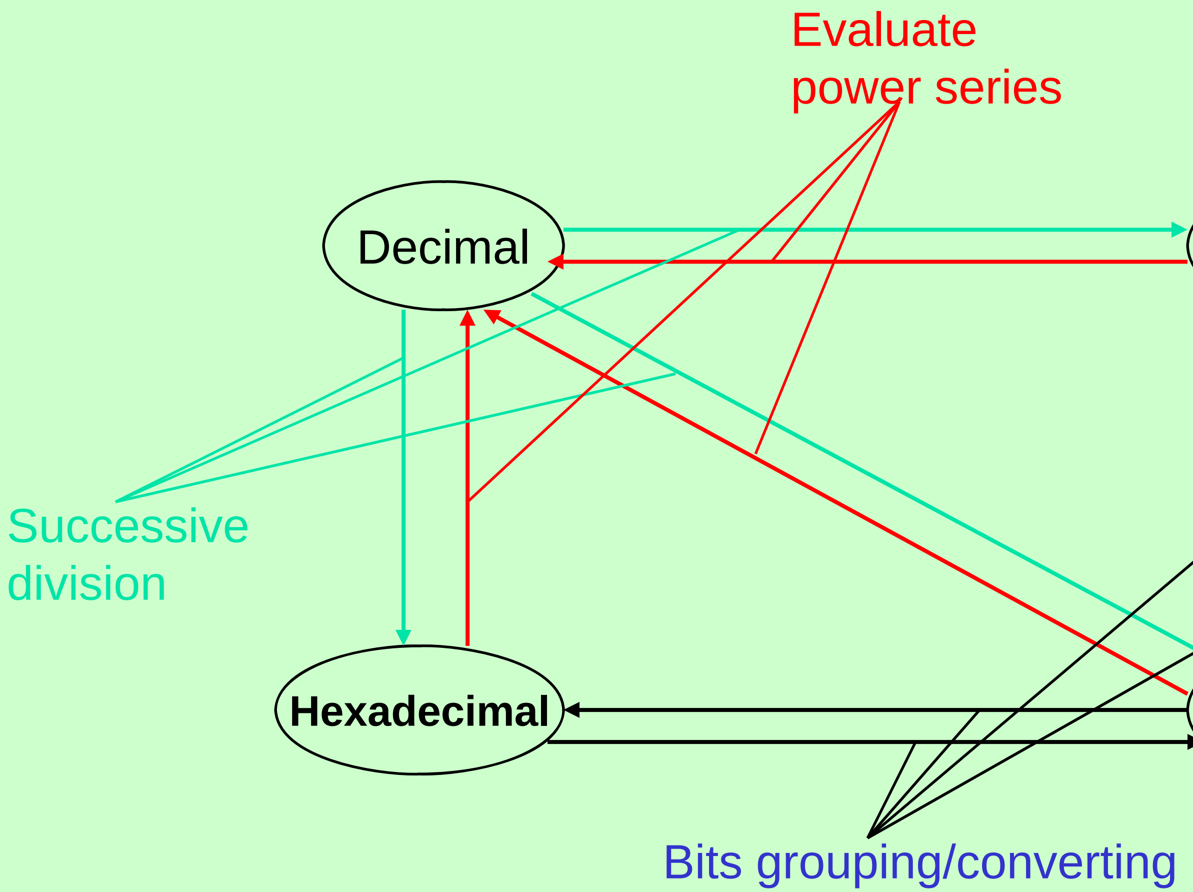
Number Systems

- Non-positional number systems
  - Ancient Egyptian base-10 number system
    - 1A1 = 11A = A11
- Positional number systems
  - $N = a_{n-1}r^{n-1} + a_{n-2}r^{n-2} + \dots + a_2r^2 + a_1r^1 + a_0$
    - n: the number of digits, r: radix/base, $0 <= a_i < r$

Number Systems

- Decimal number
  - $7642_{10} = 7 \times 10^3 + 6 \times 10^2 + 4 \times 10 + 2$
- Binary number
  - $101111_2 = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2 + 1$ $= 47_{10}$
- Octal number
  - $1352_8 = 1 \times 8^3 + 3 \times 8^2 + 5 \times 8 + 2 = 746_{10}$
- Hexadecimal number
  - 0<=a<16: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
  - $2EA_{16} = 2 \times 16^2 + 14 \times 16 + 10 = 746_{10}$

Summary of Base Conversion



Base Conversion

- Binary, Octal, Hexadecimal → Decimal
  - Evaluate the power series
- Decimal → Binary
  - Successive division by radix 2
  - Collect remainders in the reverse order
- Decimal → Octal
  - Successive division by radix 8
  - Collect remainders in the reverse order
- Decimal → Hexadecimal
  - Successive division by radix 16
  - Collect remainders in the reverse order

Base Conversion

Base-s => Base-r

INTEGER: Perform a sequence of division by r (in base-s arithmetic) and keep track of the remainders

While the fraction is not zero do:

1. Divide the base-s number/Quotient by r
2. Remainder = first coefficient of the number in base-r
3. Drop the remainder
4. Repeat from 1 until the quotient is zero

Base Conversion

FRACTION: Perform a sequence of multiplications by r (in base-s arithmetic) and keep track of the integers

1. Multiply the base-s number by r
2. Integer = first coefficient of the number in base-r
3. Drop the integer
4. Repeat from 1 until zero or reach the desired precision.

Base Conversion

2	746	
2	373	0
2	186	1
2	93	0
2	46	1
2	23	0
2	11	1
2	5	1
2	2	1
2	1	0
	0	1

$746_{10} = 1011101010_2$

8	746	
8	93	2
8	11	5
8	1	3
	0	1

$746_{10} = 1352_8$

16	746	
16	46	A
16	2	E
	0	2

$746_{10} = 2EA_{16}$

Base Conversion

- Binary ←→ Octal: group 3 bits into an octal digit, convert an octal digit into 3 bits

$$\underbrace{001}_{1}\ \underbrace{011}_{3}\ \underbrace{101}_{5}\ \underbrace{010}_{2}{}_2 = 1352_8$$

- Binary ←→ Hexadecimal: group 4 bits into a hexadecimal digit, convert a hexadecimal digit into 4 bits

$$\underbrace{0010}_{2}\ \underbrace{1110}_{E}\ \underbrace{1010}_{A}{}_2 = 2EA_{16}$$