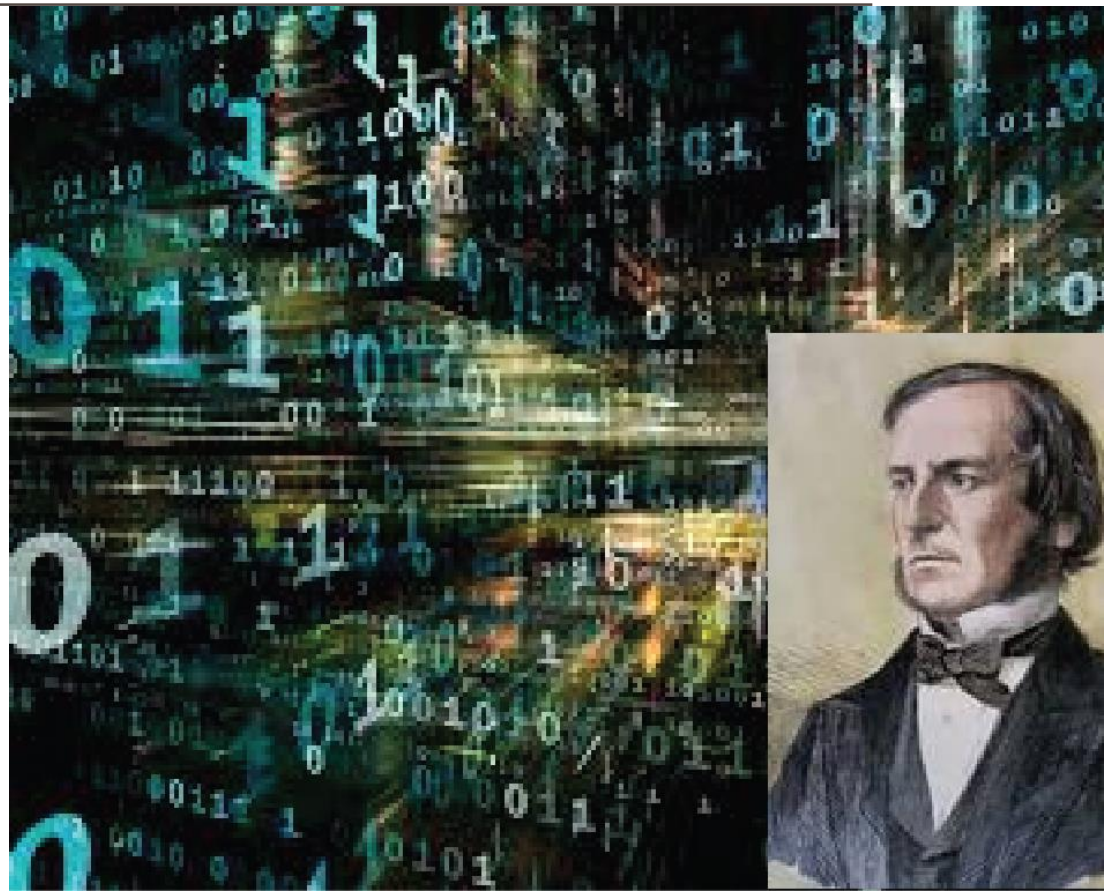
DIGITAL CIRCUITS

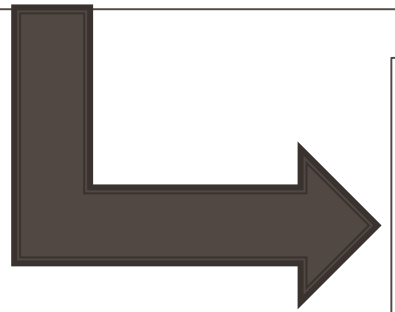
Week-10, Lecture-1
Combinational Circuits

Sneh Saurabh
9th October, 2018



Digital Circuits: Announcements/Revision

Arithmetic Operations



Multiplication

Multiplication: Signed Numbers

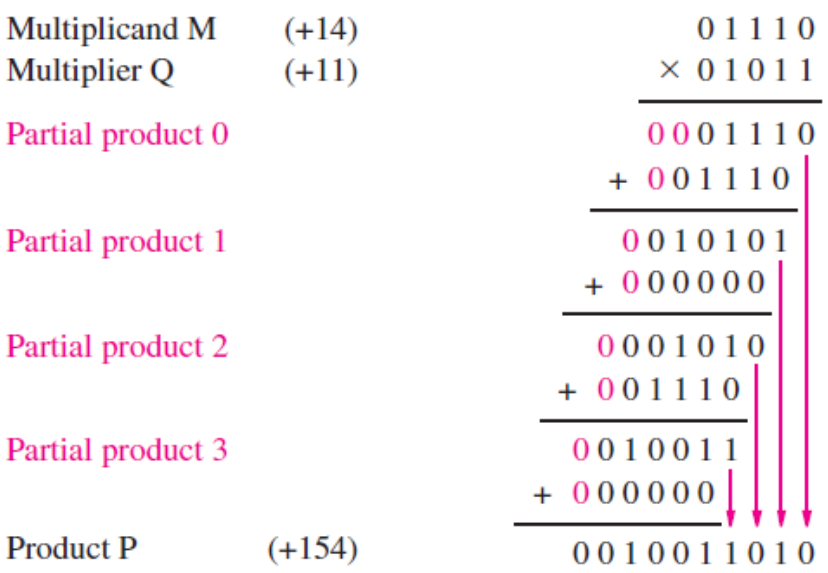
- A multiplicand can be positive or negative and multiplier is assumed to be positive

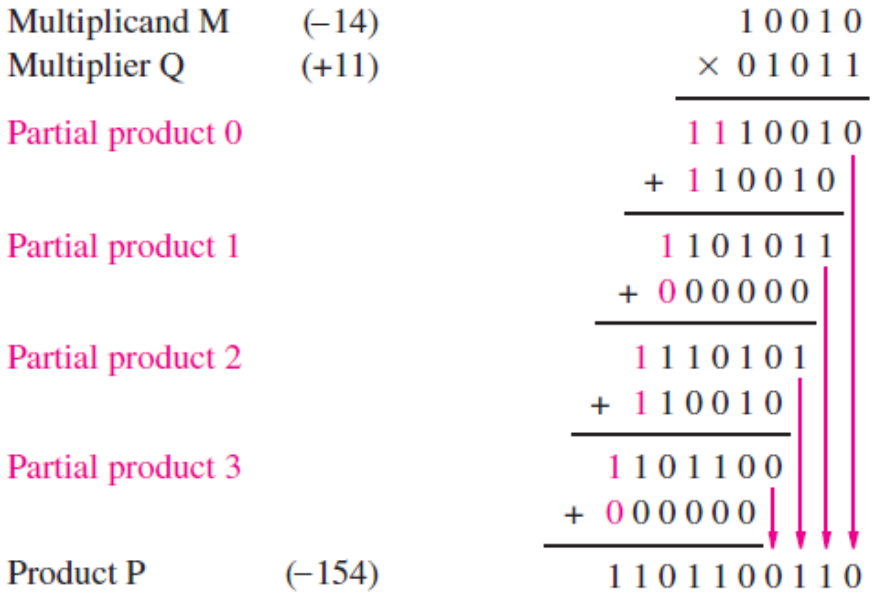
- **Similar scheme as unsigned number**: For each bit of the multiplier operand that is equal to 1, a ***properly shifted*** version of the multiplicand must be added to the partial product

- An n-bit signed number is represented as an (n + 1)-bit number by using *sign extension*, that is, by ***replicating the sign bit as the new left-most bit***.

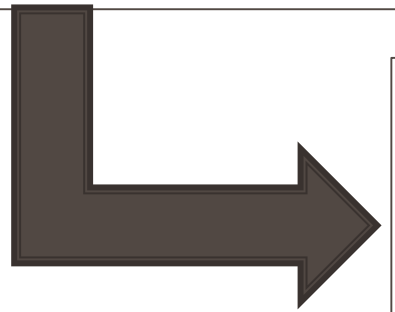
- When a shifted version of the multiplicand is added to a partial product, overflow has to be avoided. Hence the new partial product must be larger by one extra bit [This requires sign-extension]

Multiplication: Signed Numbers (Illustration)





Arithmetic Operations



BCD Addition

BCD Adder: Functionality

- Normally addition is carried out for binary digits ("0" and "1")
  - ➢ Half Adders, Full Adders

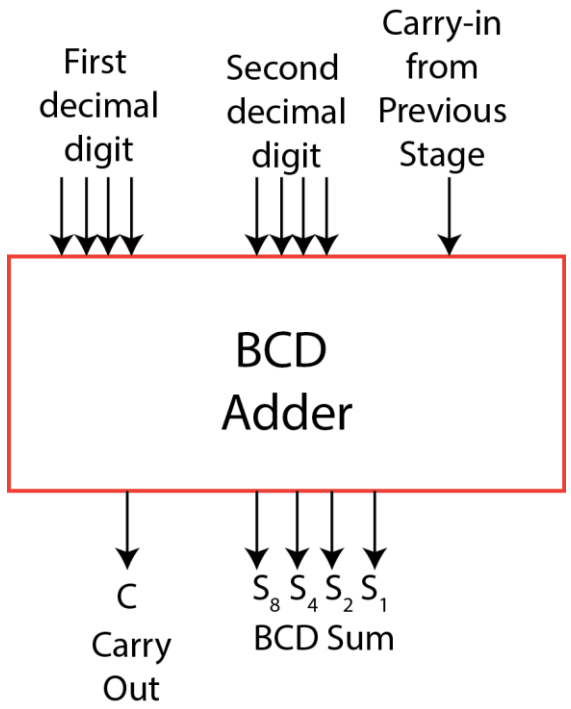
$$\begin{array}{r} 7 \\ +\ 5 \\ \hline 12 \end{array} \qquad \begin{array}{r} 8 \\ +\ 9 \\ \hline 17 \end{array}$$

Inputs

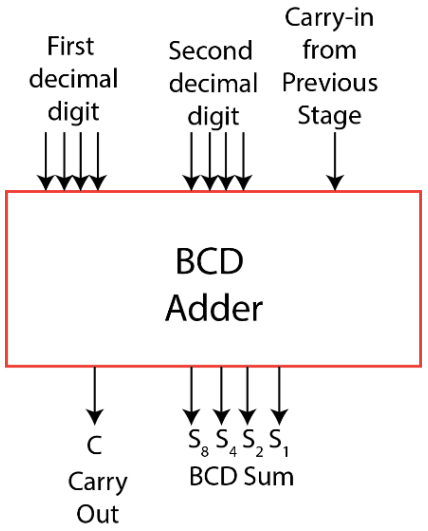
- In contrast, a BCD adder takes inputs as Decimal Number Digit (0-9)
  - ➢ Decimal Digit is coded as BCD
- It also takes an input as carry-in from previous stage (1 bit)

Outputs

- Output is the sum of input digits (S_8,S_4,S_2,S_1) in decimal notation and a carry-out (C)
  - ➢ The output sum is also coded as BCD

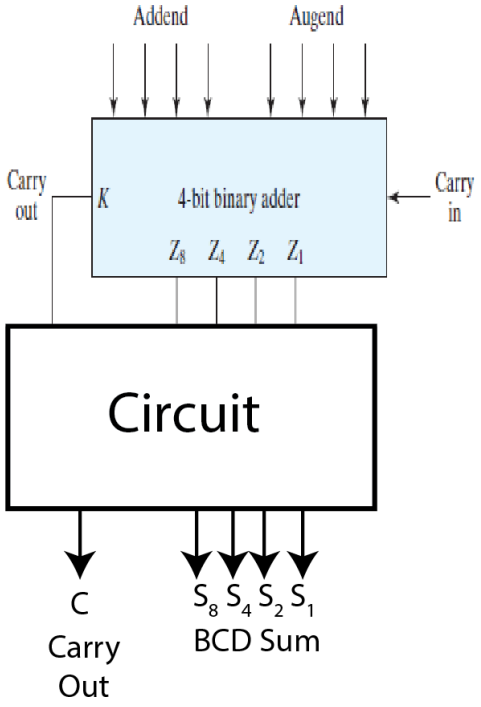


BCD Adder: Truth Table



- We want to implement BCD Adder using binary adder

- Inputs are applied to a ***four-bit binary adder***



- We need to find the truth table between output of the binary adder (K, Z_8,Z_4,Z_2,Z_1) and expected output of the BCD adder (C, S_8,S_4,S_2,S_1)

Binary Sum					BCD Sum					Decimal
K	Z_8	Z_4	Z_2	Z_1	C	S_8	S_4	S_2	S_1	
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	2
0	0	0	1	1	0	0	0	1	1	3
0	0	1	0	0	0	0	1	0	0	4
0	0	1	0	1	0	0	1	0	1	5
0	0	1	1	0	0	0	1	1	0	6
0	0	1	1	1	0	0	1	1	1	7
0	1	0	0	0	0	1	0	0	0	8
0	1	0	0	1	0	1	0	0	1	9
0	1	0	1	0	1	0	0	0	0	10
0	1	0	1	1	1	0	0	0	1	11
0	1	1	0	0	1	0	0	1	0	12
0	1	1	0	1	1	0	0	1	1	13
0	1	1	1	0	1	0	1	0	0	14
0	1	1	1	1	1	0	1	0	1	15
1	0	0	0	0	1	0	1	1	0	16
1	0	0	0	1	1	0	1	1	1	17
1	0	0	1	0	1	1	0	0	0	18
1	0	0	1	1	1	1	0	0	1	19

Rule to convert Adder Output to BCD output

Binary Sum					BCD Sum					Decimal
K	Z_8	Z_4	Z_2	Z_1	C	S_8	S_4	S_2	S_1	
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	2
0	0	0	1	1	0	0	0	1	1	3
0	0	1	0	0	0	0	1	0	0	4
0	0	1	0	1	0	0	1	0	1	5
0	0	1	1	0	0	0	1	1	0	6
0	0	1	1	1	0	0	1	1	1	7
0	1	0	0	0	0	1	0	0	0	8
0	1	0	0	1	0	1	0	0	1	9
0	1	0	1	0	1	0	0	0	0	10
0	1	0	1	1	1	0	0	0	1	11
0	1	1	0	0	1	0	0	1	0	12
0	1	1	0	1	1	0	0	1	1	13
0	1	1	1	0	1	0	1	0	0	14
0	1	1	1	1	1	0	1	0	1	15
1	0	0	0	0	1	0	1	1	0	16
1	0	0	0	1	1	0	1	1	1	17
1	0	0	1	0	1	1	0	0	0	18
1	0	0	1	1	1	1	0	0	1	19

- Output of the binary adder is in the range 0-19 (i.e. 9+9+1)
- S_8,S_4,S_2,S_1 is in the range 0-9 (a decimal digit in BCD)

Problem:

Find a rule by which the ***binary sum from the adder*** is converted to the ***BCD digit*** representation of the number in the BCD sum (i.e. the output of the BCD adder).

Solution:

- When the binary sum is equal to or less than 1001, the corresponding BCD number is identical
- When the binary sum is greater than 1001, the addition of binary 6 (0110) to the binary sum converts it to the correct BCD representation and also produces an output carry as required.

BCD Adder: Expression for carry-out

Problem: Find the logic circuit that detects the necessary condition for correction.

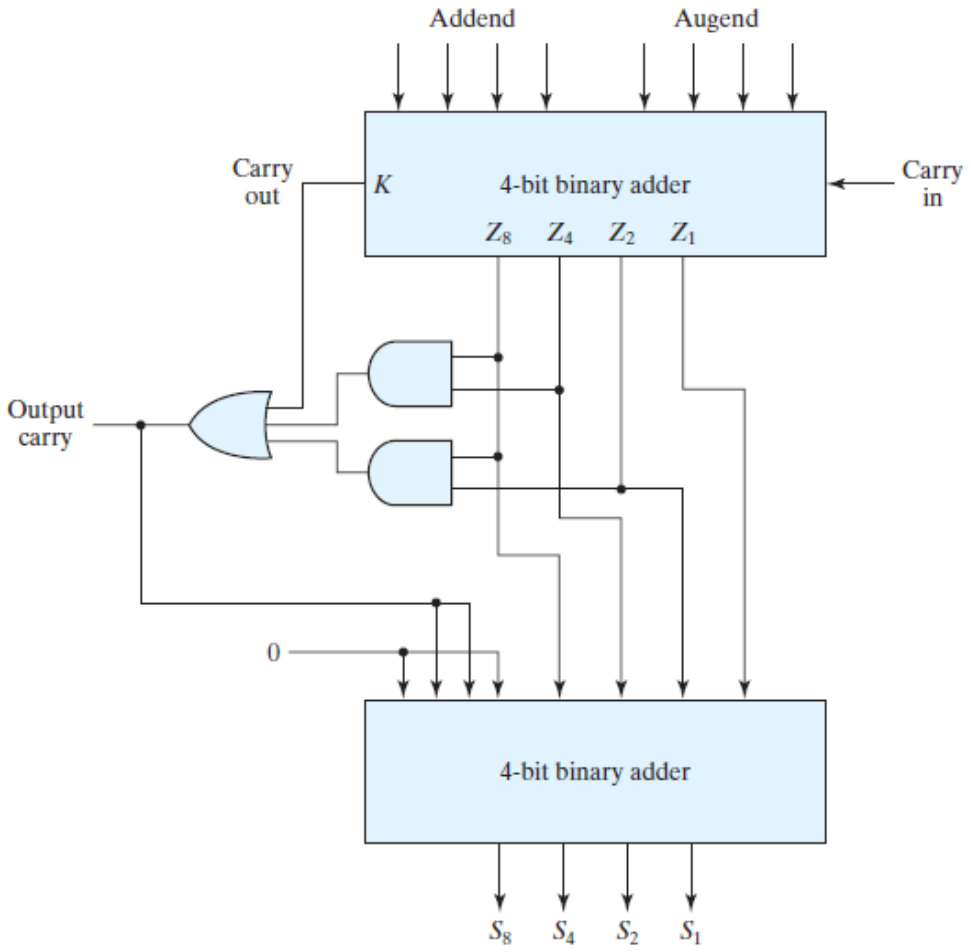
In other words, find an expression for C (carry out) of the BCD sum in terms of K, Z_8, Z_4, Z_2, Z_1

Binary Sum					BCD Sum					Decimal
K	Z_8	Z_4	Z_2	Z_1	C	S_8	S_4	S_2	S_1	
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	2
0	0	0	1	1	0	0	0	1	1	3
0	0	1	0	0	0	0	1	0	0	4
0	0	1	0	1	0	0	1	0	1	5
0	0	1	1	0	0	0	1	1	0	6
0	0	1	1	1	0	0	1	1	1	7
0	1	0	0	0	0	1	0	0	0	8
0	1	0	0	1	0	1	0	0	1	9
0	1	0	1	0	1	0	0	0	0	10
0	1	0	1	1	1	0	0	0	1	11
0	1	1	0	0	1	0	0	1	0	12
0	1	1	0	1	1	0	0	1	1	13
0	1	1	1	0	1	0	1	0	0	14
0	1	1	1	1	1	0	1	0	1	15
1	0	0	0	0	1	0	1	1	0	16
1	0	0	0	1	1	0	1	1	1	17
1	0	0	1	0	1	1	0	0	0	18
1	0	0	1	1	1	1	0	0	1	19

Solution:

- A correction is needed when the binary sum has an output carry $K = 1$.
- The other six combinations from 1010 through 1111 that need a correction have all 1 in position Z_8.
  - To distinguish them from binary 1000 and 1001,which also have a 1 in position Z_8, we specify further that either Z_4 or Z_2 must have a 1
- $C = K + Z_8.(Z_4 + Z_2)$

BCD Adder: Circuit Implementation



Top Adder

- Two decimal digits, together with the input carry, are added in the top four-bit adder to produce the ***binary sum***

Bottom Adder

- When the output carry is equal to 0, nothing is added to the binary sum.
- When the output carry is equal to 1, binary 0110 is added to the binary sum through the bottom four-bit adder.
- The carry out of bottom adder can be ignored, since it supplies information already available at the output carry terminal.

BCD Adder: Multiple digits

- Addition of multiple decimal digits (N) needs N BCD adder stages
- The output carry from one stage must be connected to the input carry of the next higher order stage.

Problem: Build the circuit for BCD addition of two numbers having two decimal digits. For example

$(24)_{10}+(32)_{10}= (56)_{10}$

$(0010\ 0100)_{BCD}+(0011\ 0010)_{BCD}= (0101\ 0110)_{BCD}$

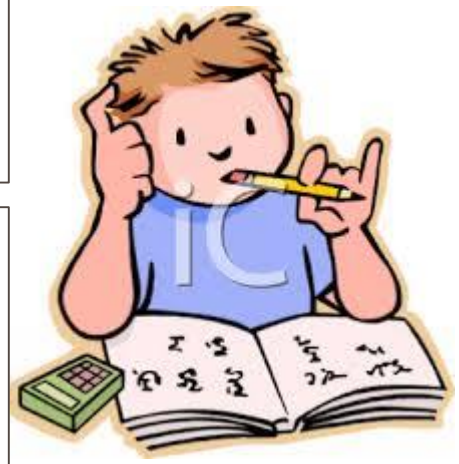
Digital Circuits: Practice Problems

Problems 4.18-4.19

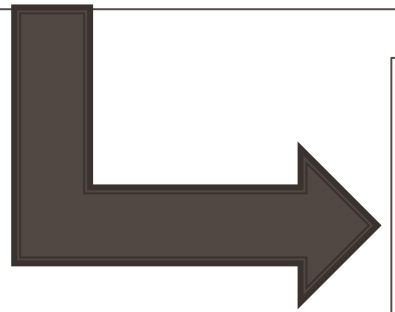
from “Digital Design”– M. Morris Mano & Michael D. Ciletti, Ed-5, Pearson (Prentice-Hall).

Problems 5.18, 5.19

from Fundamentals of Digital Logic with Verilog Design - S. Brown, Z. Vranesic



Digital Circuits

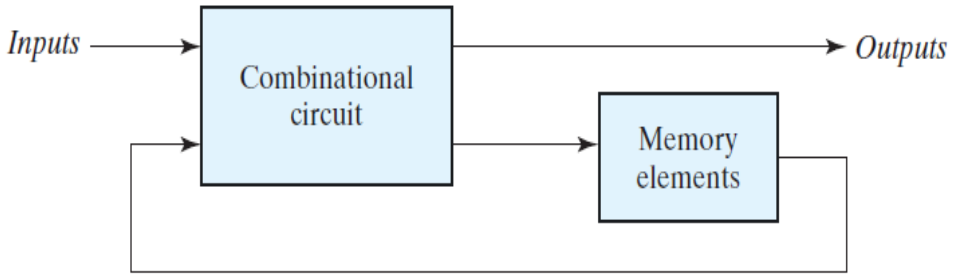


Sequential Circuits

Sequential Circuit: Introduction

Combinational Circuit	Sequential Circuit
Output depends only on present input	Output depends not only on present input, but also on past sequence of inputs
Do not have memory	Have memory
Do not have feedback	Have feedback
Examples: AND/OR gates, MUX, Decoder, Adder, Multiplier	Flip-flops and latches

Sequential Circuit: Block Diagram



- Combinational and memory elements are connected to form ***a feedback path***.
- Memory element capable of storing binary information(0 or 1).
- The binary information (0 or 1) in memory elements define the ***state of the sequential circuit*** at that time.

- Outputs in a sequential circuit are a function not only of the ***inputs***, but also of the ***present state*** of the storage elements
- The ***next state*** of the storage elements is also a function of ***external inputs*** and the ***present state***
- Thus, a sequential circuit is specified by a time sequence of inputs, outputs, and internal states