VHDL Introduction

VHDL is an acronym for Very high speed integrated circuit (VHSIC)Hardware Description Language which is a programming language that describes a logic circuit by function, data flow behavior, and/or structure.

This hardware description is used to configure a programmable logic device (PLD), such as a field programmable gate array (FPGA), with a custom logic design.

Structure of a VHDL File

The general format of a VHDL program is built around the concept of BLOCKS which are the basic building units of a VHDL design. A digital system in VHDL consists of a design entity which can contain other entities. Each entity is modeled by an Entity declaration and an Architecture body.

1. **Entity Declaration:** A VHDL design begins with an ENTITY block that describes the interface for the design. It defines the names, input and output logic signals of the circuit being designed. Its syntax is shown below:

```
entity entity_name is
        Port declaration
end entity_name;
```

2. **Architecture:** The ARCHITECTURE block describes the internal operation of the design. Within these blocks are numerous other functional blocks used to build the design elements of the logic circuit being created. Its syntax is as shown below:

```
architecture architecture_name of entity_name
architecture_declarative_part;

begin
        Statements;
end architecture_name;
```

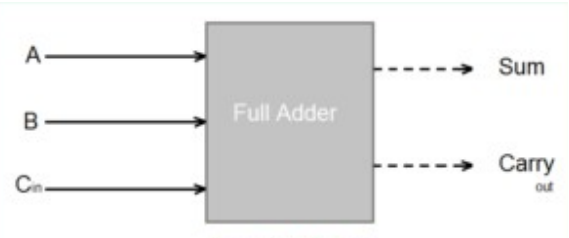
Problem Statement 1: Design of a 1-Bit Full Adder

Introduction:

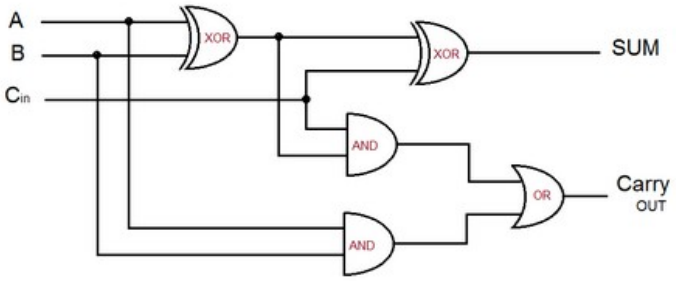
An Adder is a digital electronic circuit that performs addition of numbers. A 1-Bit Full Adder is a logical circuit that performs an addition operation on three binary digits C_{in}, A and B and outputs two 1-bit binary digits which are Sum(S) and Carry Out(C_{out}). A full adder is a Combinational Circuit. That is, a circuit whose output is dependent only on the state of its inputs.

Design Of a 1-Bit Full Adder:

Block Diagram



Combinational Circuit



Truth Table

INPUTS			OUTPUTS	
A	B	C_{in}	SUM	CARRY OUT
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

Boolean Expressions after simplifying the Karnaugh Map derived from the truth table:

$Sum = A \oplus B \oplus C_{in}$

$Carry\ out = A.B + (A \oplus B).C_{in}$

These boolean expressions are used to design the circuit above.

VHDL Code:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

Entity Full_adder is
        Port (
         A : in STD_LOGIC;
         B : in STD_LOGIC;
         Cin : in STD_LOGIC;
         S : out STD_LOGIC;
         Cout : out STD_LOGIC);
End Full_adder;

Architecture test of Full_adder is
Begin
        S <= A XOR B XOR Cin ;
        Cout <= (A AND B) OR (Cin AND A) OR (Cin AND B) ;
End test
```

Problem Statement 2: Design of a Simple Washing Machine

Introduction:

A Washing Machine is a Sequential Machine. That is, it is a machine that operates using a sequential circuit.

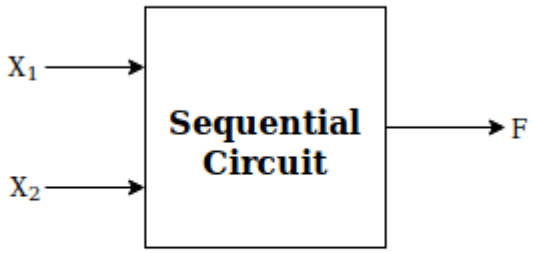
Sequential circuit: is a circuit whose outputs depend not only on its inputs, but also on the present state of system. Sequential logic systems are Finite State Machines (FSMs).

Finite State Machines: consist of a set of states, some inputs, some outputs, and a set of rules for moving from state to state.

In order to design the Washing Machine, we must design its FSM first. Once the FSM is fully designed, we can easily write out the design in VHDL.

Design Of a Simple Washing Machine:

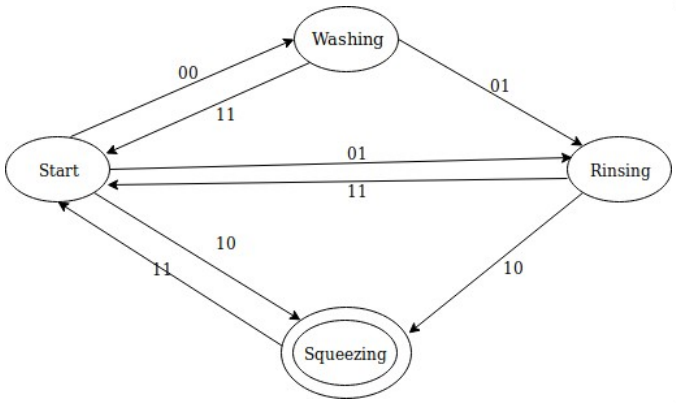
Block Diagram



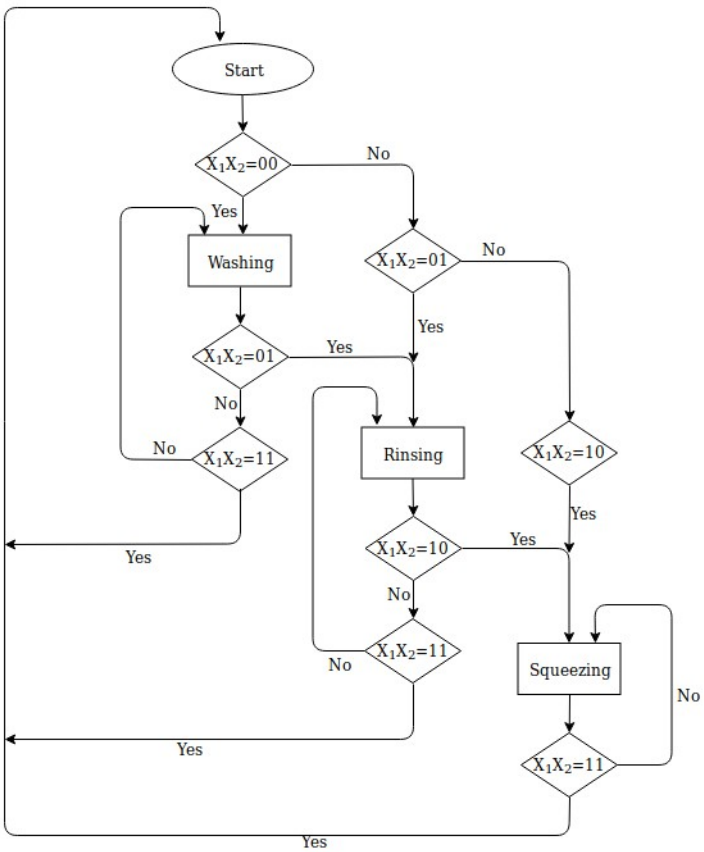
State Table

Present State	Input X_1X_2	Next State
Start	00	Washing
Start	01	Rinsing
Start	10	Squeezing
Washing	01	Rinsing
Washing	11	Start
Rinsing	10	Squeezing
Rinsing	11	Start
Squeezing	11	Start

State Machine Diagram:



State Machine chart:



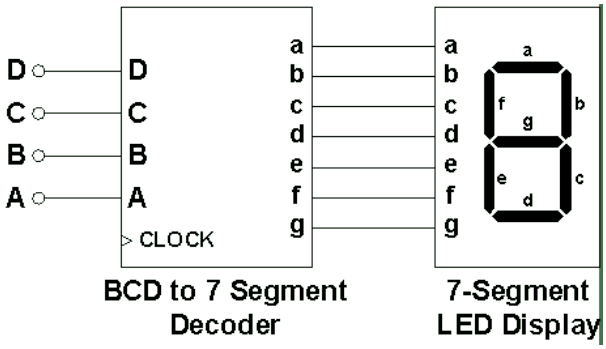
Problem Statement 3: BCD to 7 Segment Display Decoder

Introduction:

A BCD to seven segment decoder has four input lines (A, B, C and D) and 7 output lines (a, b, c, d, e, f and g), this output is given to seven segment LED display which displays the decimal number depending upon inputs.

Design Of a BCD to 7 Segment Display Decoder:

Block Diagram



Truth Table

Decimal Digit	Input lines				Output lines							Display pattern
	A	B	C	D	a	b	c	d	e	f	g	
0	0	0	0	0	1	1	1	1	1	1	0	0
1	0	0	0	1	0	1	1	0	0	0	0	1
2	0	0	1	0	1	1	0	1	1	0	1	2
3	0	0	1	1	1	1	1	1	0	0	1	3
4	0	1	0	0	0	1	1	0	0	1	1	4
5	0	1	0	1	1	0	1	1	0	1	1	5
6	0	1	1	0	1	0	1	1	1	1	1	6
7	0	1	1	1	1	1	1	0	0	0	0	7
8	1	0	0	0	1	1	1	1	1	1	1	8
9	1	0	0	1	1	1	1	1	0	1	1	9

Outputs:

a = F1 (A, B, C, D) = ∑m (0, 2, 3, 5, 7, 8, 9)

b = F2 (A, B, C, D) = ∑m (0, 1, 2, 3, 4, 7, 8, 9)

c = F3 (A, B, C, D) = ∑m (0, 1, 3, 4, 5, 6, 7, 8, 9)

d = F4 (A, B, C, D) = ∑m (0, 2, 3, 5, 6, 8)

e = F5 (A, B, C, D) = ∑m (0, 2, 6, 8)

f = F6 (A, B, C, D) = ∑m (0, 4, 5, 6, 8, 9)

g = F7 (A, B, C, D) = ∑m (2, 3, 4, 5, 6, 8, 9)

After the simplification of the Boolean Functions:

$$a = A + C + BD + \bar{B}\,\bar{D}$$

$$b = \bar{B} + \bar{C}\,\bar{D} + CD$$

$$c = B + \bar{C} + D$$

$$d = \bar{B}\,\bar{D} + C\,\bar{D} + B\,\bar{C}\,D + \bar{B}\,C + A$$

$$e = \bar{B}\,\bar{D} + C\,\bar{D}$$

$$f = A + \bar{C}\,\bar{D} + B\,\bar{C} + B\,\bar{D}$$

$$g = A + B\,\bar{C} + \bar{B}\,C + C\,\bar{D}$$

VHDL Code:

```
Library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

Entity BCD_7SDD is
  Port (
    X: in std_logic_Vector(3 DOWNTO 0);
    F: out std_logic_Vector(6 DOWNTO 0)
  );
End BCD_7SDD;

--X3-A X2-B X1-C X0-D

Architecture Behv of BCD_7SDD is
Begin
  F(6) <= X(3) OR X(1) OR (X(2) AND X(0)) OR (NOT X(2) AND NOT X(0));  --A
  F(5) <= (NOT X(2)) OR (NOT X(1) AND NOT X(0)) OR (X(1) AND X(0));  --B
  F(4)<= X(2) OR NOT X(1) OR X(0);--C

  F(3)<= (NOT X(2) AND NOT X(0)) OR (X(1) AND NOT X(0)) OR (X(2) AND NOT X(1) AND X(0)) OR (NOT
X(2) AND X(1)) OR X(3);--D

  F(2)<= (NOT X(2) AND NOT X(0)) OR (X(1) AND NOT X(0));--E
  F(1)<= X(3) OR (NOT X(1) AND NOT X(0)) OR (X(2) AND NOT X(1)) OR (X(2) AND NOT X(0));--F
  F(0)<= X(3) OR (X(2) AND NOT X(1)) OR ( NOT X(2) AND X(1)) OR (X(1) AND NOT X(0));--G
End Behv;
```