ABSTRACT
In this paper, we present a novel approach to design a digital circuit using VHDL language. Our technique for designing digital systems totally based on VHDL. It's like a C language.

key words: Digital Circuit, VHDL, Entity, Architecture.

INTRODUCTION
VHDL means VHSIC Hardware Description Language where VHSIC stands for Very-High-Speed Integrated Circuit. VHDL is a hardware description language that can be used to model a digital system. VHDL is commonly used as a design-entry language for field-programmable gate arrays and application-specific integrated circuits in electronic design automation of digital circuits.

1> HISTORY
The requirements for the language were first generated in 1981 under the VHSIC program. In this program, a number of U.S. companies were involved in designing VHSIC chips for the Department of Defense (DOD). At that time, most of the companies were using different hardware description languages to describe and develop their integrated circuits. As a result, different vendors could not effectively exchange designs with one another. Also, different vendors provided DOD with descriptions of their chips in different hardware description languages. Reprocurement and reuse was also a big issue. Thus, the need for a standardized hardware description language for design, documentation, and verification of digital systems was generated.

VHDL was originally developed at the behest of the US Department of Defense in order to document the behavior of the ASICs that supplier companies were including in equipment. That is to say, VHDL was developed as an alternative to huge, complex manuals which were subject to implementation-specific details. In addition to IEEE standard 1164, several child standards were introduced to extend functionality of the language. IEEE standard 1076.2 added better handling of real and complex data types. IEEE standard 1076.3 introduced signed and unsigned types to facilitate arithmetical operations on vectors. IEEE standard 1076.1 (known as VHDL-AMS) provided analog and mixed-signal circuit design extensions. Some other standards support wider use of VHDL, notably VITAL (VHDL Initiative towards ASIC Libraries) and microwave circuit design extensions.

2> CAPABILITIES
a> The language can be used as an exchange medium between chip vendors and CAD tool users. Different chip vendors can provide VHDL descriptions of their components to system designers. CAD tool users can use it to capture the behavior of the design at a high level of abstraction for functional simulation.

b> The language can also be used as a communication medium between different CAD and CAE tools, for example, a schematic capture program may be used to generate a VHDL description for the design which can be used as an input to a simulation program.

c> The language supports hierarchy, that is, a digital system can be modeled as a set of interconnected components; each component, in turn, can be modeled as a set of interconnected subcomponents.

d> The language supports flexible design methodologies: top-down, bottom-up, or mixed.

e> The language is not technology-specific, but is capable of supporting technology-specific features. It can also support various hardware technologies, for example, you may define new logic types and new components, you may also specify technology-specific attributes. By being technology independent, the same behavior
model can be synthesized into different vendor libraries.

> It supports both synchronous and asynchronous timing models

### 3> Basic Structure of a VHDL file

VHDL is used to describe a model for a digital hardware device. This model specifies the external view of the device and one or more internal views. The internal view of the device specifies the functionality or structure, while the external view specifies the interface of the device through which it communicates with the other models in its environment.

### 4> BASIC TERMINOLOGY

VHDL is a hardware description language that can be used to model a digital system. The digital system can be as simple as a logic gate or as complex as a complete electronic system. A hardware abstraction of this digital system is called an entity in this text. An entity X, when used in another entity Y, becomes a component for the entity Y. Therefore, a component is also an entity, depending on the level at which you are trying to model. To describe an entity, VHDL provides five different types of primary constructs, called design units.

They are

- a. Entity declaration
- b. Architecture body
- c. Configuration declaration
- d. Package declaration
- e. Package body

**a. ENTITY DECLARATION**

The entity declaration defines the name of the entity and lists the input and output ports. The general form is as follows,

```vhdl
entity NAME_OF_ENTITY is [ generic
generic_declarations);]
end [NAME_OF_ENTITY];
```

**b. ARCHITECTURE BODY**

The internal details of an entity are specified by an architecture body using any of the following modeling styles:

1. As a set of interconnected components (to represent structure),
2. As a set of concurrent assignment statements (to represent dataflow)
3. As a set of sequential assignment statements (to represent behavior).
4. Any combination of the above three.

The architecture body specifies how the circuit operates and how it is implemented. As discussed earlier, an entity or circuit can be specified in a variety of ways, such as behavioral, structural (interconnected components), or a combination of the above.

The architecture body looks as follows,

```vhdl
architecture architecture_name of NAME_OF_ENTITY is
-- Declarations
-- components declarations
-- signal declarations
-- constant declarations
-- function declarations
-- procedure declarations
-- type declaration
begin
-- Statements
end architecture_name;
```

**c. CONFIGURATION DECLARATION**

A configuration declaration is used to select one of the possibly many architecture bodies that an entity may have, and to bind components, used to represent structure in that architecture body, to entities represented by an entity architecture pair or by a configuration, that reside in a design library.

**d. PACKAGE DECLARATION**

A package declaration is used to store a set of common declarations like components,
types, procedures, and functions. These declarations can then be imported into other design units using a context clause.

e. PACKAGE BODY
A package body is primarily used to store the definitions of functions and procedures that were declared in the corresponding package declaration, and also the complete constant declarations for any deferred constants that appear in the package declaration. Therefore, a package body is always associated with a package declaration; furthermore, a package declaration can have at most one package body associated with it. Contrast this with an architecture body and an entity declaration where multiple architecture bodies may be associated with a single entity declaration.

EXAMPLES

1. **D-FF WITH AN ENABLE INPUT**

```vhdl
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY d_ff IS
    PORT (R, RESETN, E, CLK: IN STD_LOGIC;
        Q: OUT STD_LOGIC);
END d_ff;
ARCHITECTURE behavior OF d_ff IS
begin
    PROCESS (RESETN, CLK,R,E)
    BEGIN
        IF RESETN = '0' THEN
            Q <= '0';
        ELSIF CLK'EVENT AND CLK = '1' THEN
            IF E = '1' THEN
                Q <= R;
            ELSE
                Q <= Q;
            END IF;
        END IF;
    END PROCESS;
END behavior;
```

**FULL ADDER**

```vhdl
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY full_adder IS
    PORT (a, b, c: IN STD_LOGIC;
        sum, cout: OUT STD_LOGIC);
END full_adder;
ARCHITECTURE adder OF full_adder IS
begin
    sum <= a XOR b XOR c;
    cout <= (a AND b) OR (a AND c) OR (b AND c);
END adder;
```

CONCLUSION
Using VHDL programming we can design any circuit from gate level to processor level. It is easy to learn and high capability with software.

REFERENCES
1. AVHDL primer by J. Bhasker
2. VHDL programming by example by Douglas L.Perry
3. Circuit design with VHDL by Volnei A. Pedroni
4. VHDL Modular Design and Synthesis of cores and system by Zainalabedin Navabi.