FACULTY OF ELECTRONIC ENGINEERING & TECHNOLOGY

NMJ20404/EKT221

DIGITAL ELECTRONICS II

LAB 2

DIGITAL LOGIC DESIGN USING QUARTUS

LAB 2: DIGITAL LOGIC DESIGN USING QUARTUS

INTRODUCTIONS

Digital electronic circuit designs can be classified into combinational logic and sequential logic circuits.

Combinational logic circuits consist of logic gates whose outputs at any time are determined by combining the values of the applied inputs using logic operations. A combinational circuit performs an operation that can be specified logically by a set of Boolean expressions. It consists of input variables, output variables, logic gates and interconnections. The interconnected logic gates accept signals from the inputs and generate signals at the outputs.

By referring to the block diagram in Figure 1.1, the *m* input variables come from the environment of the circuit, and the *n* output variables are available for use by the environment. Each input and output variable exists physically as a signal that represents logic '1' or logic '0'.

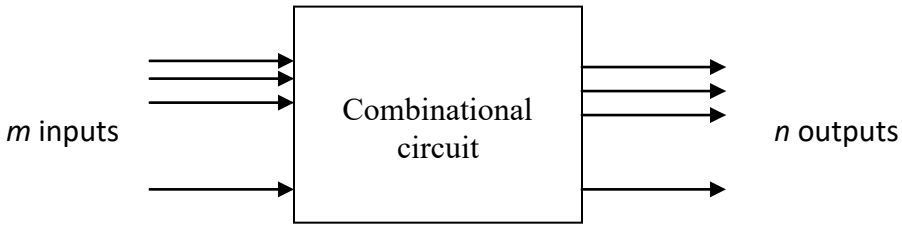

Figure 1.1: Combinational Circuit Block Diagram

Combinational circuit design procedure by using Quartus may include the following steps:

1	Specification definition	Define the specification for your design.
2	Formulation derivation	Derive the truth table/Boolean equations between the inputs and outputs.
3	Level optimization	Apply two-level or multiple-level optimization.
4	Verification	Functional or timing simulation.

Sequential logic circuits are combinatorial system with some of the outputs fed back as inputs. This makes the digital circuit perform a "sequence" of operations. The simplest sequential system is probably a flip flop, a mechanism that represents a binary digit or "bit". Sequential logic differs from combinational logic in that the output of the system is dependent not only on the present inputs to the device, but also on past inputs; *i.e.*, the output of a sequential logic device depends on its present internal state and the present inputs. This implies that a sequential logic device has some kind of *memory* of at least part of its “history” (*i.e.*, its previous inputs).

Sequential systems are often designed as state machines. Figure 1.2 describes a block diagram of a sequential circuit which combines both the combinational and the storage elements.

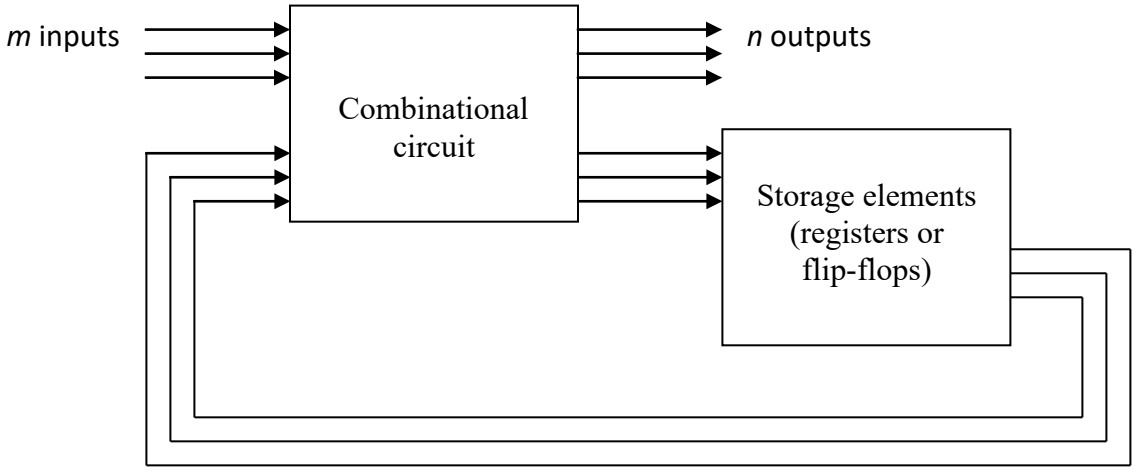

Figure 1.2: Sequential Circuit Block Diagram

Sequential circuit design procedure by using Quartus may include the following steps:

1	Specification definition	Define the specification for your design.
2	Formulation derivation	Derive the state table/diagram.
3	State assignment	Transfer the table/diagram into binary codes/bit level.
4	Flip-flop determination	Determine the choice of flip-flops (storage elements) for the system.
5	Output equation determination	Derive output equations from the state table (if any).
6	Level optimization	Apply two-level or multiple-level optimization.
7	Verification	Functional or timing simulation.

TASK / ASSIGNMENT

Write a Verilog code for each of the following digital circuits. Then, compile and simulate the code to verify the circuit design.

1. A majority function is a system that will assert when there are more 1's than 0's on the inputs. Design a 4-bit majority function.

2. Design a 3-bit up and down counter that has an enable input X. When X=0, the counter will start counting down the value. When X=1, the counter continues to count up the value.

3. Figure 1.3 represents a *relative-magnitude detector* that takes two 2-bit binary numbers, and determines the operations as follow:
   a. when A = B ; P = HIGH
   b. when A > B ; Q = HIGH
   c. when A < B ; R = HIGH

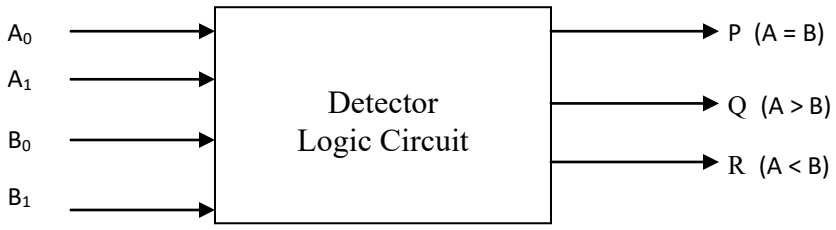

Figure 1.3