shathenk

HALIBEC048

chapter 1: Analysis of clocked sequential circuits

* some flip-flops have asynchronous i/p that are used to force the flip-flop to a particular state independently of the clock.

* The i/p that sets the flip-flop to 1 called preset or direct set. The input that clears the flip-flop to 0 is called clear or direct reset.

* when power is turned on in a digital system, the state of the flip-flop is unknown. The direct i/p are useful for bringing all flip-flops in the system to known starting state prior to the clocked op^n.

* The info available in a state table can be represented graphically in the form of state diagram. In this type of diagram a state is represented by a circle & the (clock-triggered) transitions b/n states are indicated by directed lines connecting the circles

* The time sequence of i/p, o/p & flip-flop states can be enumerated in a state table (transition table). The table has 4 parts present state, next state, i/p & o/p.

* In general a sequential ckt with 'm' flip-flops & 'n' i/p needs 2^{m+n} rows in the state table.

chapter-2: Analysis with D-flip-flops.

* The i/p eqn of D-flip flop is given by $D_A = A \oplus x \oplus y$. D_A means a D flip-flop with o/p A.

* The x & y variables are the i/p to the ckt. No o/p eqn are given which implies that the o/p comes from the o/p of the flip-flop.

* The state table has one column for the present state of flip-flop 'A' two clms for the two i/p, & one colomn for the next state of A.

* The next-state values are obtained from the state eqn $A(t+1) = A \oplus x \oplus y$.

* The eqn specifies an odd func^n & it equal to 1 when one variable is 1 or when all 3 variables are 1.

Phython:

chapter 1: object oriented programming.

① object oriented pgm^g explained

② turning this appln into oop style

turning this appln into oop style part 2

③ creating a bank account object

④ Inheritance

⑤ oop glossary

⑥ GUI in oop design (practice)

⑦ solution.