

Python:

App 5: Build a desktop database app ⑧ desktop
to build a pgm database app- how the O/P will look like

→ How to build a pgm which is used store the information of books.

→ user interface design

→ build a graphical url in interface

→ formatted interface

we use grid method make thing easier

① from tkinter import

② window TK()

③ window main loop()

l = label (window, text, title)

l = grid (row = 0, column = 0)

copy for more row & column indicating "title",

"author", "year", title 'text' - showing var()

e = entry (window, text variable, title text)

e = grid (row = 0, column = 1)

This should be applied for author, year, to get a row & column table.

→ backend: we got to know to create table ⑨
create books.

→ connecting the frontend to the backend

→ fixing the bug

→ creating a standard executable version pgm

Logic design:

boolean algebra:

'0' & '1'

→ cost of the ckt

→ simple realization of a ckt

→ In 1864, George Boole developed an algebra system called boolean algebra.

→ boolean algebra is a system of mathematical logic.

→ It is designed with set of elements, a set of operators, & a no of axioms & postulates.

→ set of element (0,1)

→ two binary operators - OR & AND

→ unary operator - Not (Invert)

• Axioms & laws of boolean algebra: -

→ Axioms & postulates of boolean algebra are a logical expn upon which we can build digital theories.

"AND opn" OR opn"

"NOT opn"

$$\left. \begin{array}{l} 0 \cdot 0 = 0 \\ 0 \cdot 1 = 0 \\ 1 \cdot 0 = 0 \\ 1 \cdot 1 = 1 \end{array} \right\} \text{AND opn}$$

$$\left. \begin{array}{l} 0 + 0 = 0 \\ 0 + 1 = 1 \\ 1 + 0 = 1 \\ 1 + 1 = 1 \end{array} \right\} \text{OR opn}$$

$$\rightarrow \bar{0} \times 0 = 1 \quad \text{or } A \times \bar{A}$$

$$\rightarrow 1 \times 1 = 0$$

Diff b/n boolean algebra, ordinary algebra & binary system.

→ In boolean algebra

$$\begin{aligned} A + A &= A & A \cdot A &= A \\ 1 + 1 &= 1 & 1 \cdot 1 &= 1 \end{aligned}$$

→ In ordinary algebra

$$\begin{aligned} A + A &= 2A & A \cdot A &= A^2 \\ 1 + 1 &= 2 & 1 \cdot 1 &= 1 \end{aligned}$$

→ In binary system

$$\begin{aligned} 1 + 1 &= (1 \cdot 0) & 1 \cdot 1 &= 1 \end{aligned}$$

Axioms postulated

$$\rightarrow x + 0 = x$$

$$x - 0 = x$$

$$\rightarrow x + 1 = 1$$

$$x \cdot 1 = x$$

$$\rightarrow x + x = x$$

$$x \cdot x = x$$

$$\rightarrow x + \bar{x} = 1$$

$$x \cdot \bar{x} = 0$$

$$\rightarrow x \cdot (\bar{x})' = x$$

• Identify elements

OR op

AND op

additive identity = 0 = $x + 0 = x$

$$x \cdot 1 = x \Rightarrow 1 = x$$

multiplicative identity = 1

$$x + 0 = x$$

$$0 + x = x$$

laws of boolean algebra:

(i) commutative: $x + y = y + x$

$$A + B = B + A \quad \text{or} \quad x \cdot y = y \cdot x$$

$$x \cdot y = y \cdot x$$

(ii) Associative law:

$$x + (y + z) = (x + y) + z$$

$$x'(y \cdot z) = (x \cdot y)' \cdot z$$

$$A + (B + C) = (A + B) + C$$

$$A(B \cdot C) = (A \cdot B) \cdot C$$

(iii) Distributive law

$$① x(y + z) = xy + xz$$

$$A=0, B=1, C=0$$

$$\text{LHS } A(B + C) = 0(1 + 0) = 0$$

$$A(B + C) = AB + AC$$

$$\text{RHS } 0B + 0C = 0$$

$$② x + yz = (x + y)(x + z)$$

$$A + BC = (A + B)(A + C)$$

$$① x + yz = (x + y)(x + z)$$

$$= x \cdot x + xz + xy + yz$$

$$= x + xz + xy + yz$$

$$= x(1 + z + y) = x \cdot 1 = x \Rightarrow x + yz \text{ hence parallel}$$

Theorem of Boolean algebra:

① Absorption theorem: $x + xy = x$, $A + AB = A$

$$\rightarrow x(1 + y) = x \cdot 1 = x$$

$$② x + \bar{x}y = x + y$$

$$x + \bar{x}(x + y) = 1 \cdot (x + y) = x + y$$

$$③ A + BC = (A + B)(A + C)$$

$$= A + BC$$

map to logic gates:

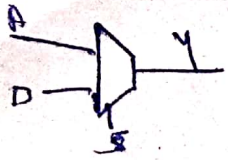
① NAND, NOR - universal gates

② universal logic

decoder are called "universal logic"

$$2^n - 1 \Rightarrow 2^n \rightarrow \text{i/p}$$

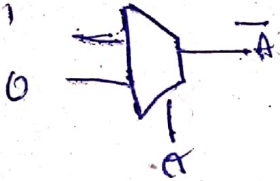
$n \rightarrow$ selection lines



S	O/P (Y)
0	A
1	B

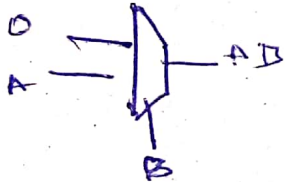
$$\Rightarrow Y = A\bar{S} + BS$$

inverted design help of mux

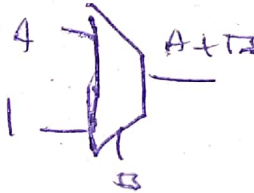


$$Y = A\bar{S} + BS \Rightarrow \bar{A} + 0A$$

$$\underline{Y = \bar{A}}$$

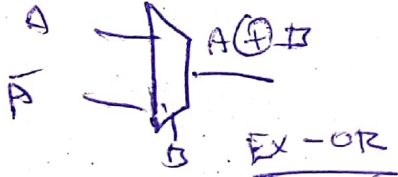


$$Y = B\bar{S} + AS$$

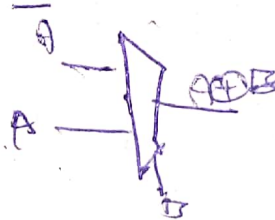


$$Y = A+B$$

$$Y = A \cdot B$$

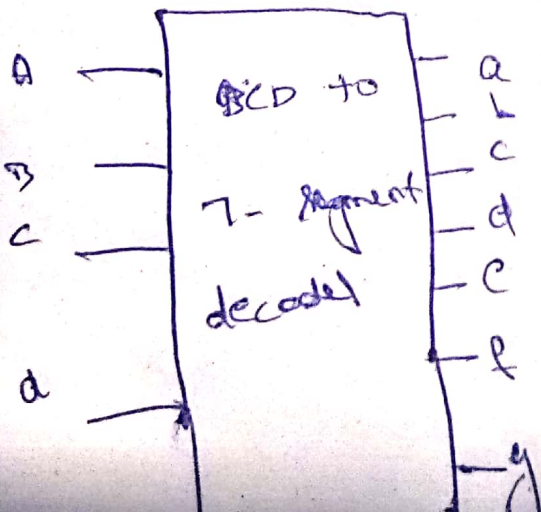


EX-OR



EX-OR

BCD to 7 segment decoder:



4 i/p lines & 7 O/P lines

