Field **P**rogrammable **G**ate **A**rrays

Field Programmable Gate Arrays (FPGAs)

- Field programmable gate array is a VLSI module that can be programmed to implement a digital system consisting of tens of thousands of gates.

A Field Programmable Gate Array *(FPGA)*

is similar to a PLD, but whereas PLDs are generally limited to hundreds of gates, FPGAs support thousands of gates. They are especially popular for prototyping integrated circuit designs. Once the design is set, hardwired chips are produced for faster performance

Field Programmable Gate Arrays (FPGAs) are divided into two major categories:

1. SRAM-based FPGAs
2. Antifuse-based FPGAs

some of the antifuse switches.

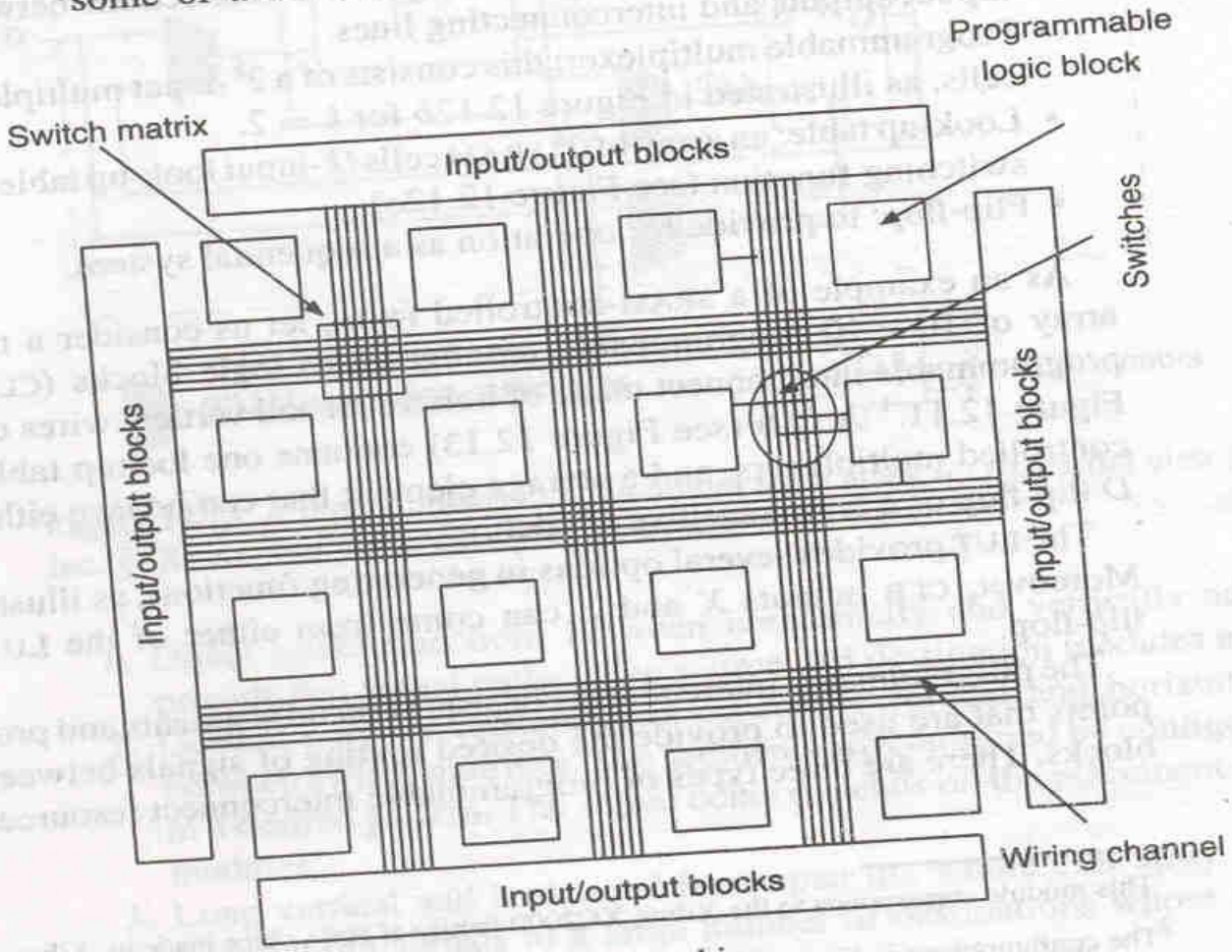
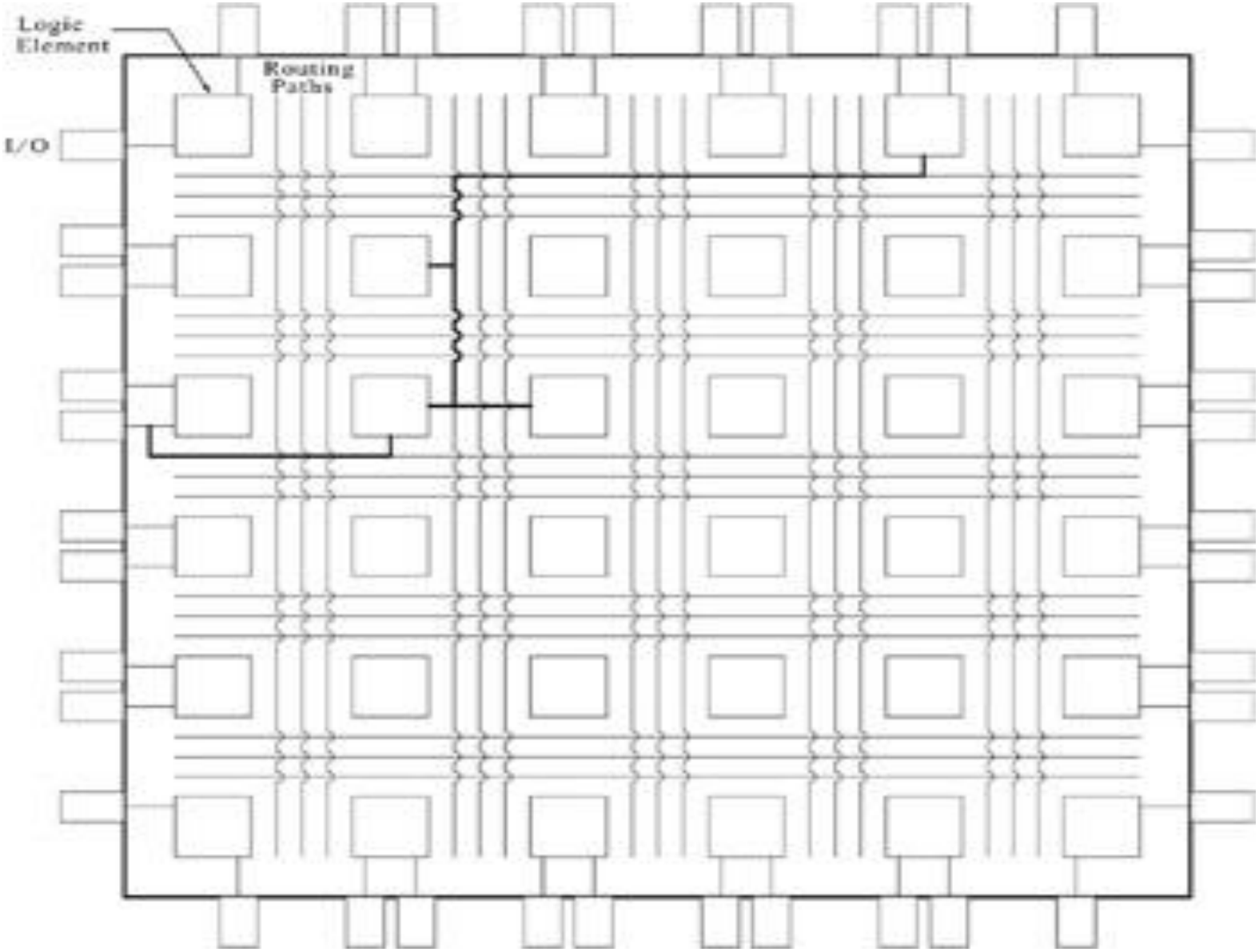


Figure 12.11 Organization of an FPGA chip.


Logic
Element
Routing
Paths
I/O

Field Programmable Gate Arrays (FPGAs)

- Field programmable gate array is a VLSI module that can be programmed to implement a digital system consisting of tens of thousands of gates.

- An **FPGA** consists of an array of three kinds of programmable devices.
  . **Logic blocks**, either combinational and or sequential
  . **Interconnection points** (switches) and
  . **Input/output blocks**

In addition there are wires grouped in vertical and horizontal channels.

Each logic modules can be programmed to implement several switching functions

Logic modules used in FPGA are used as look-up tables (LUTs) or multiplexers

On chip latches (memory cells) that are set with bit patterns to define the chip configuration, called SRAM-FPGA

Volatile, programming information not preserved after chip powered down.

Memory cells are loaded during programming Phase with binary values that represent the desired value of control signals that define chip configuration

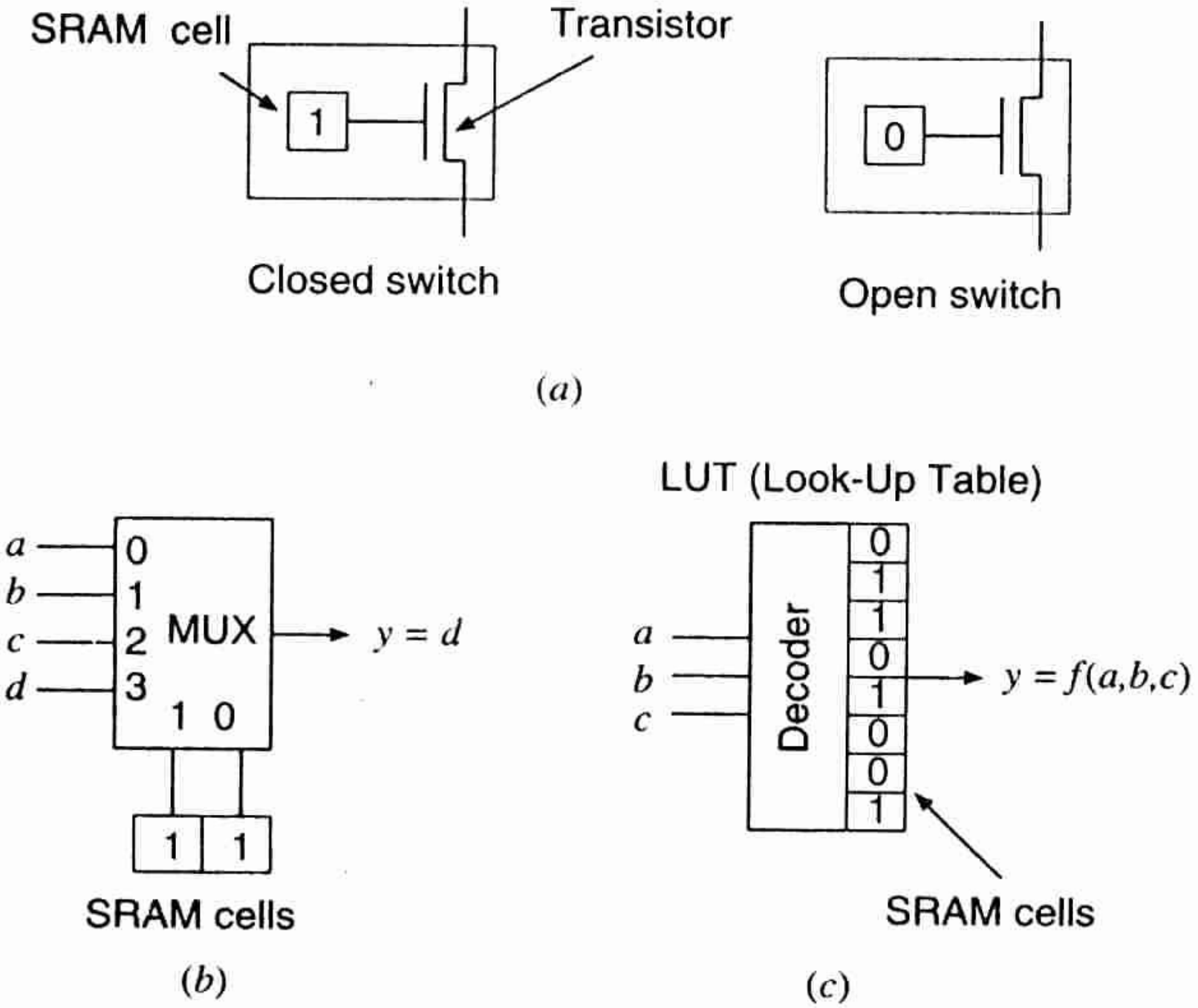
SRAM- FPGA

-Programmable switch

-Programmable Multiplexer

-Look-up table

-Flip-flop

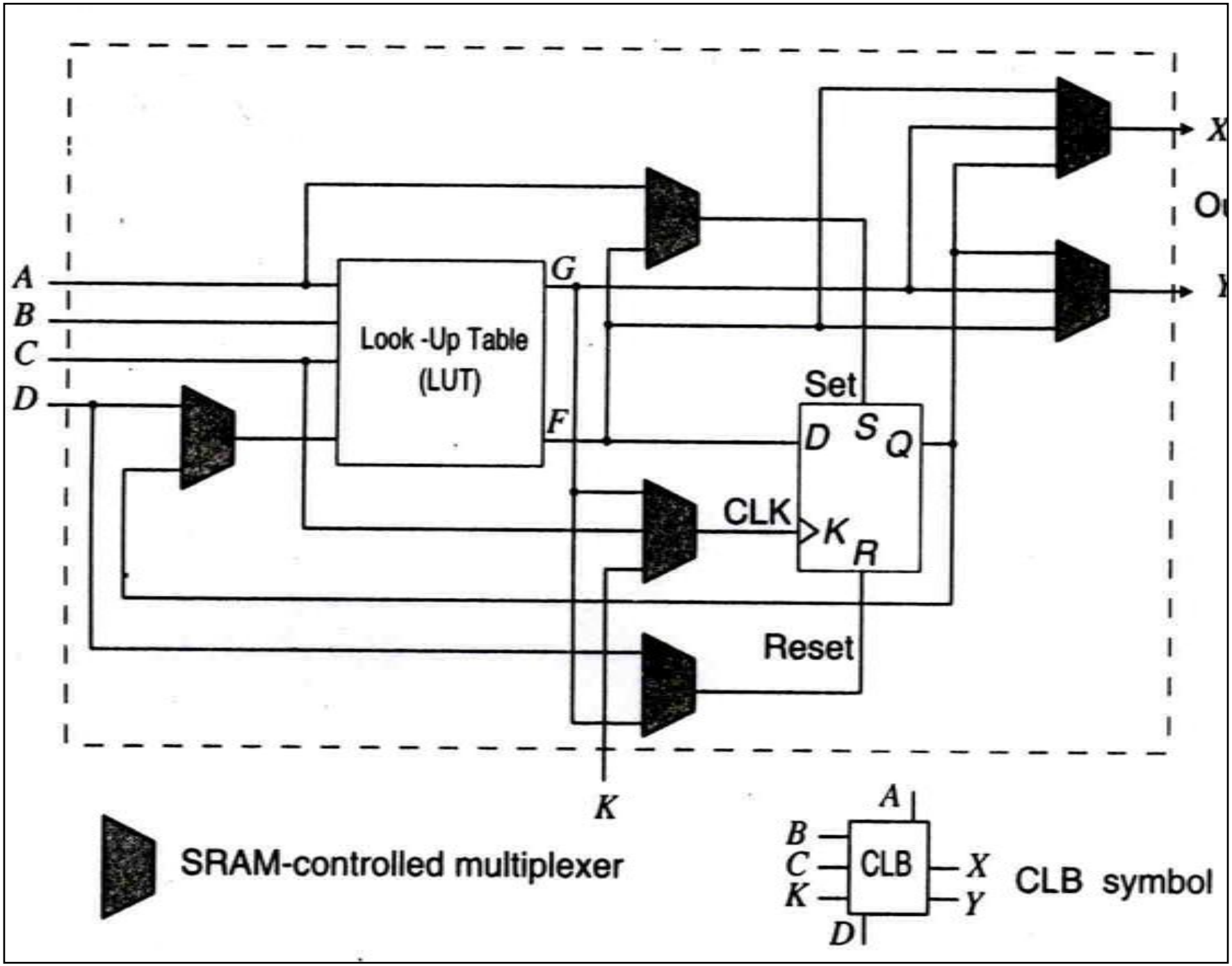

SRAM cell
Transistor
1
Closed switch
0
Open switch
(a)
a
b
c
d
0
1
2
3
MUX
1 0
y = d
1
1
SRAM cells
(b)
LUT (Look-Up Table)
a
b
c
Decoder
0
1
1
0
1
0
0
1
y = f(a,b,c)
SRAM cells
(c)

A Configurable Logic Block (CLB)

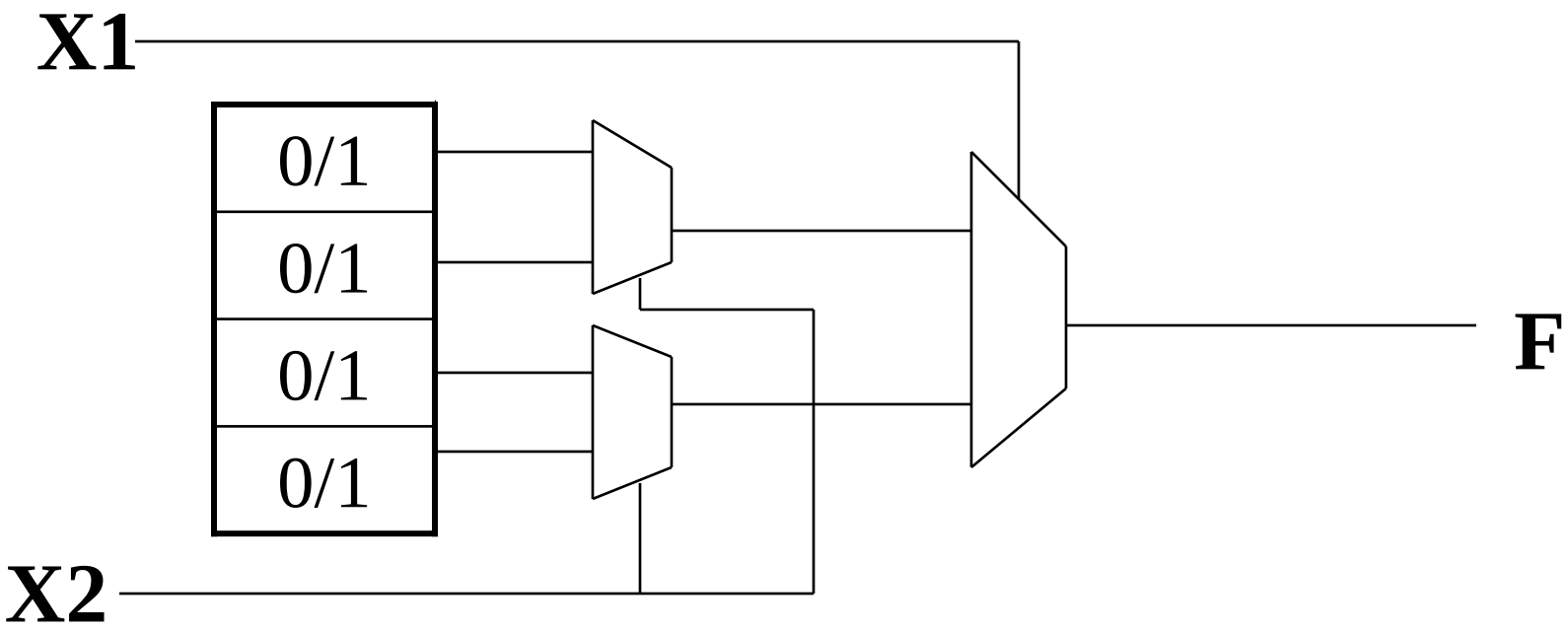
-These are programmable blocks .

-May consists of LUTs,several Multiplexers controlled by memory cells, Flip-flops

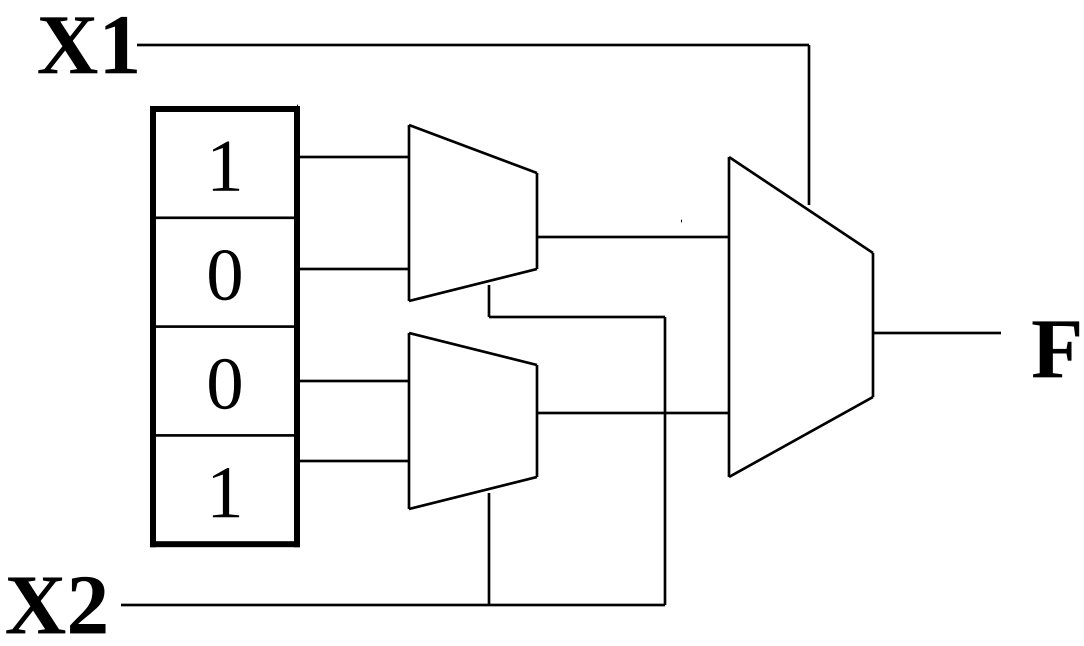
- Provide outputs.


A
B
C
D
Look -Up Table
(LUT)
G
F
Set
D
S
Q
CLK
K
R
Reset
K
X
SRAM-controlled multiplexer
A
B
C
K
D
CLB
X
Y
CLB symbol

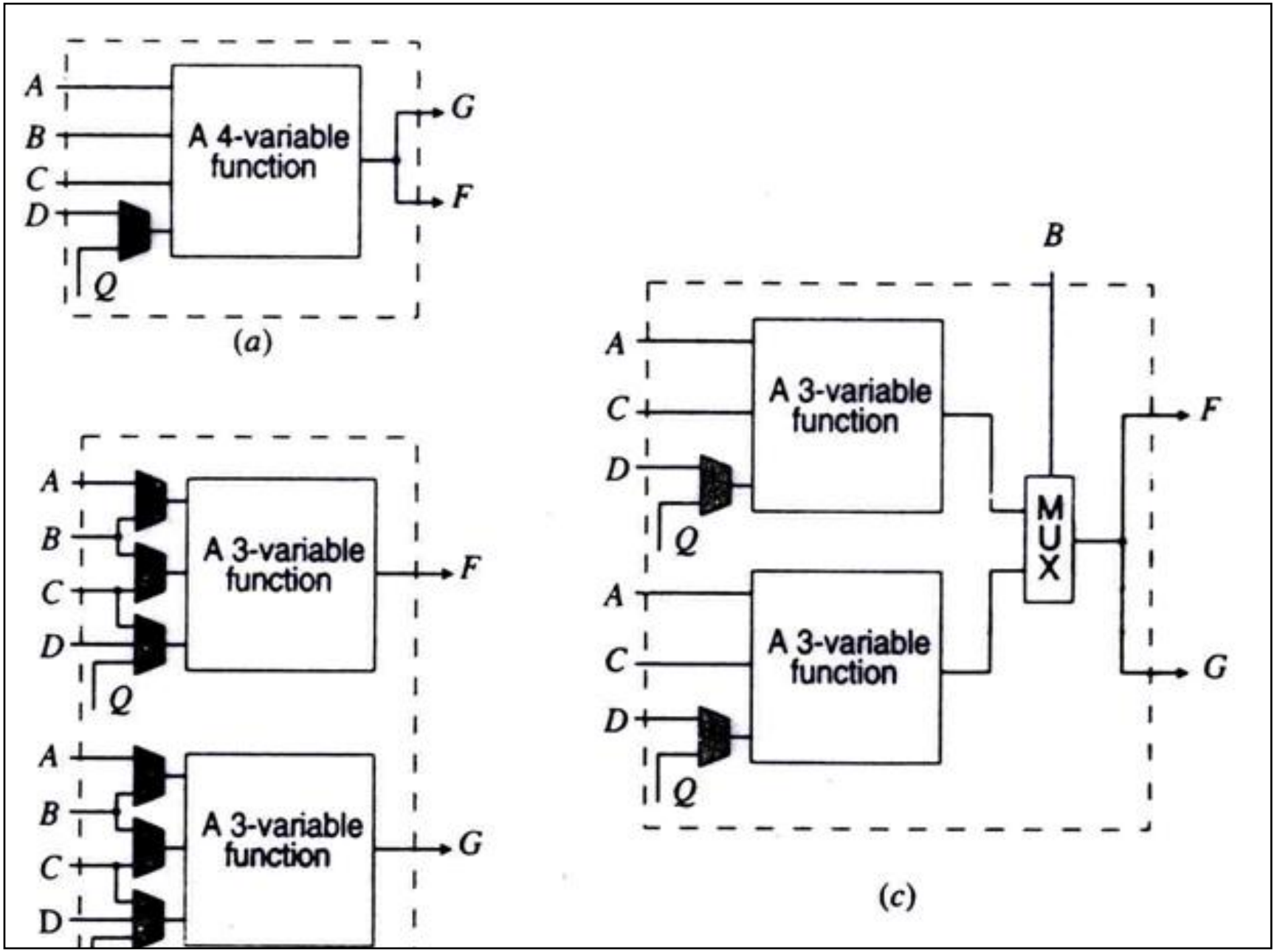
LOOK UP TABLE



LOOK UP TABLE



X1	X2	F
0	0	1
0	1	0
1	0	0
1	1	1


A
B
C
D
Q
A 4-variable function
G
F
(a)
A
B
C
D
Q
A 3-variable function
F
A
B
C
D
Q
A 3-variable function
G
B
A
C
D
Q
A 3-variable function
A
C
D
Q
A 3-variable function
MUX
F
G
(c)

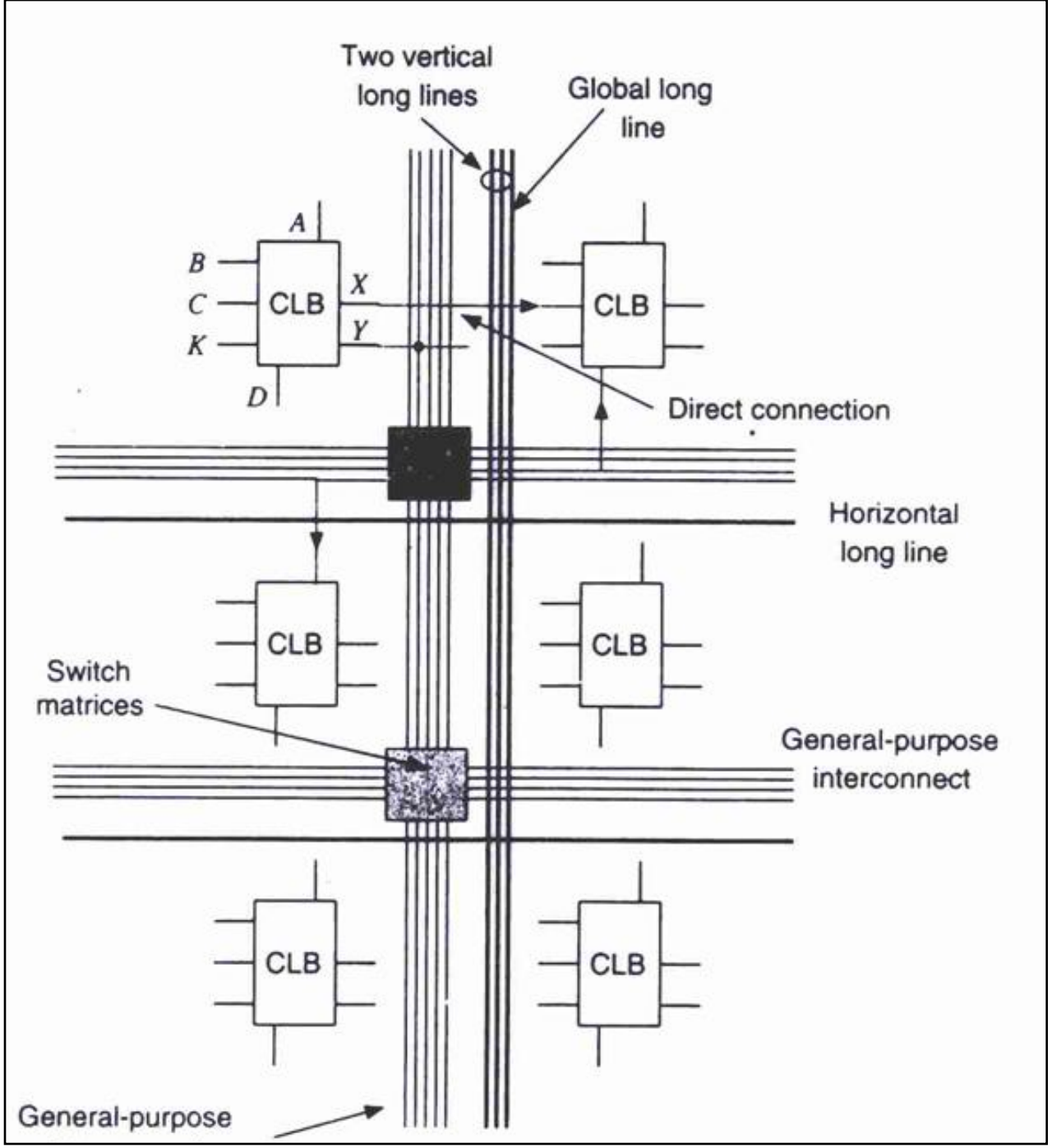
LUTs

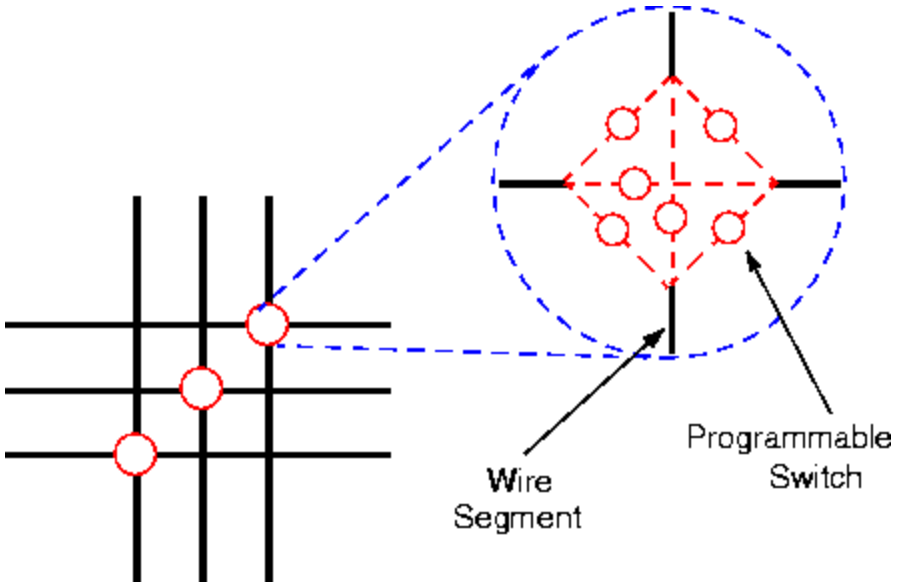
Programmable interconnects consists of metal segments And programmable switch points used to provide routing of signals

-Direct interconnections between horizontal and vertical CLBs

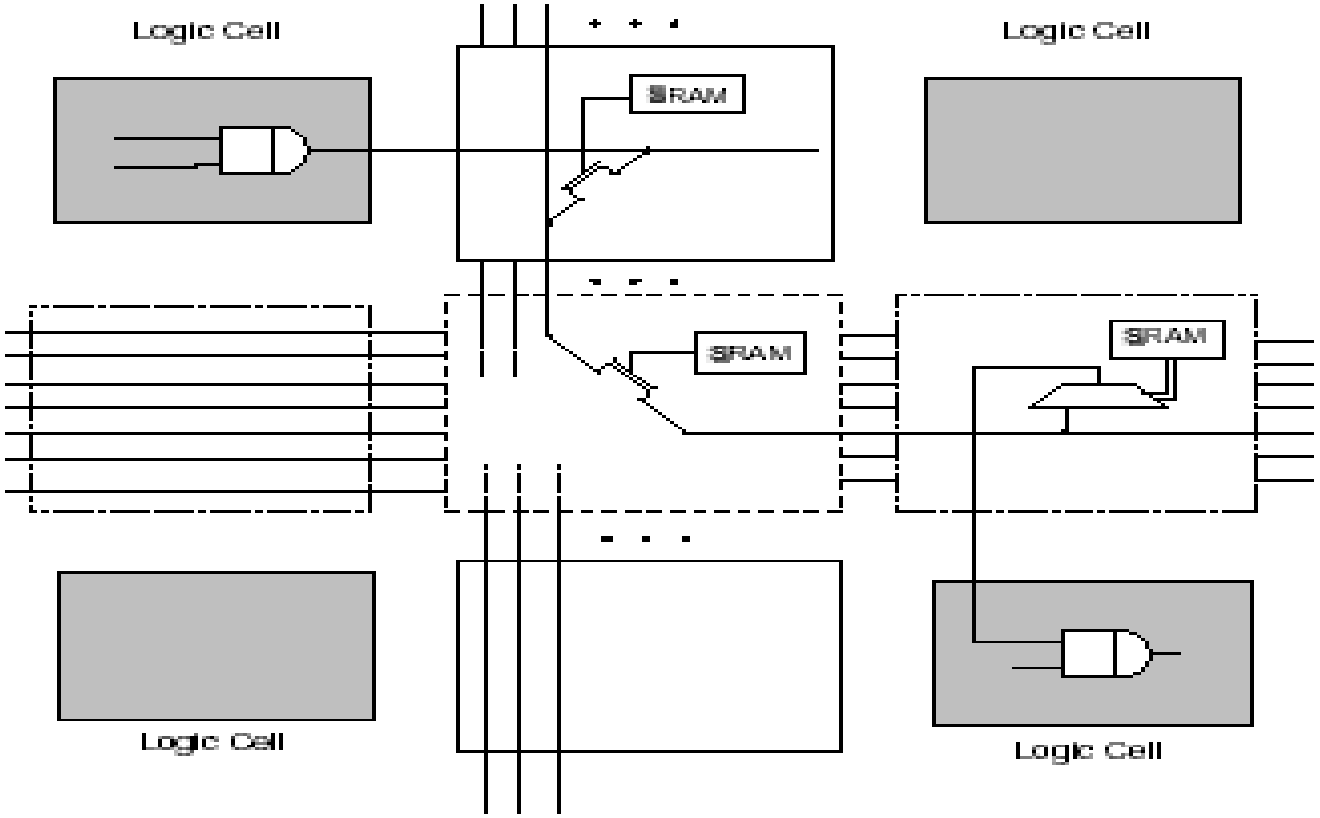
-General purpose interconnects consists of vertical and horizontal wiring segments between switch matrices.

-Long vertical and horizontal lines span the whole CLB array , providing means for transmitting signals to large number of destinations.

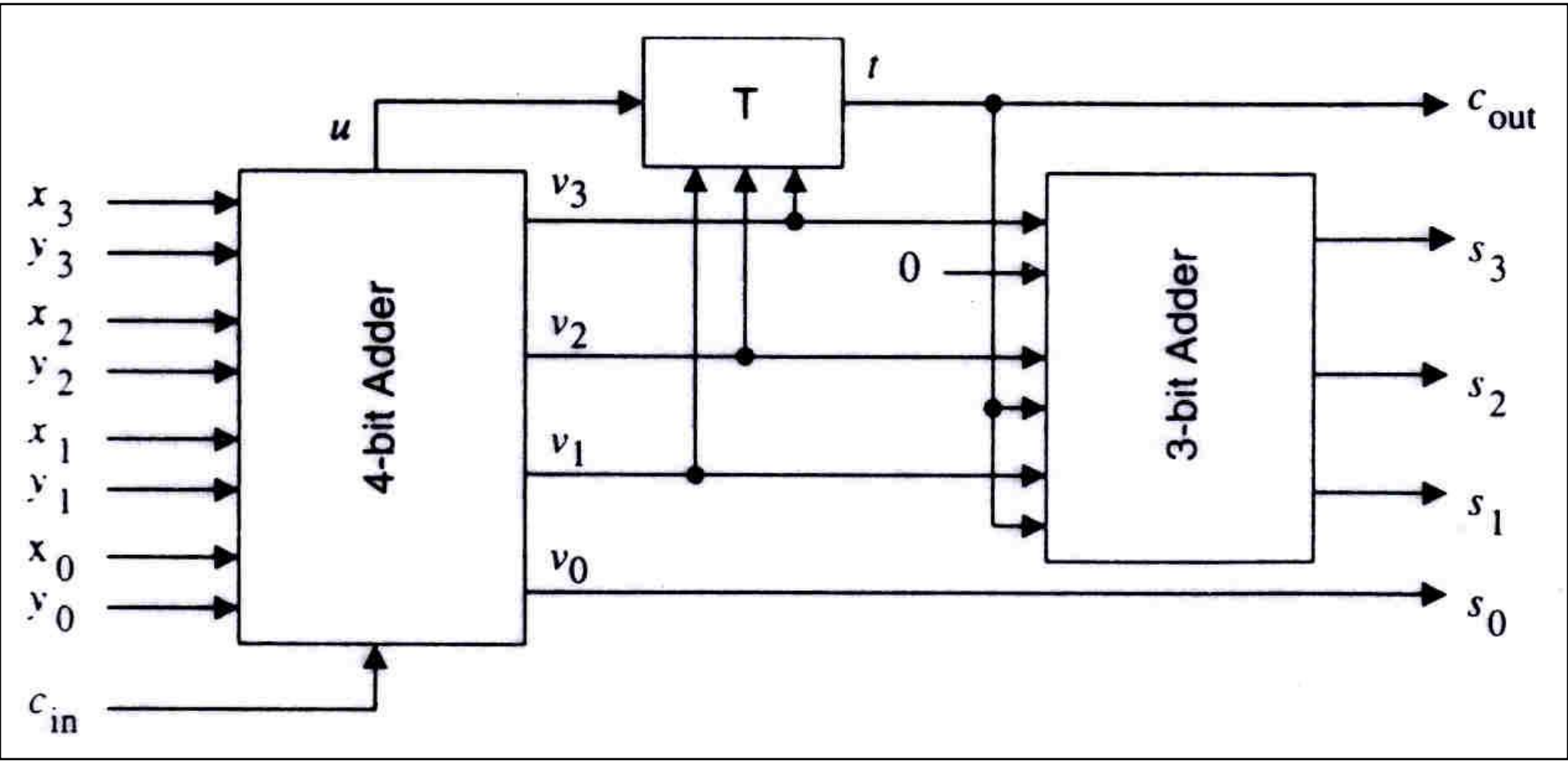

Two vertical long lines
Global long line
A
B
C
K
CLB
X
Y
D
CLB
Direct connection
Horizontal long line
CLB
CLB
Switch matrices
General-purpose interconnect
CLB
CLB
General-purpose


Wire
Segment
Programmable
Switch

SRAM CONTROLLED SWITCH



- One-Digit BCD Adder



One digit BCD adder

- The system has nine inputs and five outputs
- Performing a modulo16 addition and then a correction factor
- First adder produces

$v = (x+y+c_{in}) \bmod 16$

$u = 1$ if $(x+y+c_{in}) >= 16$

$= 0$ otherwise

T = 1 if u = 1 or v > = 10

= 0 otherwise

System function is

S = (x+y+c_{in}) mod 10

Cout = 1 if (x+y+c_{in}) > = 10

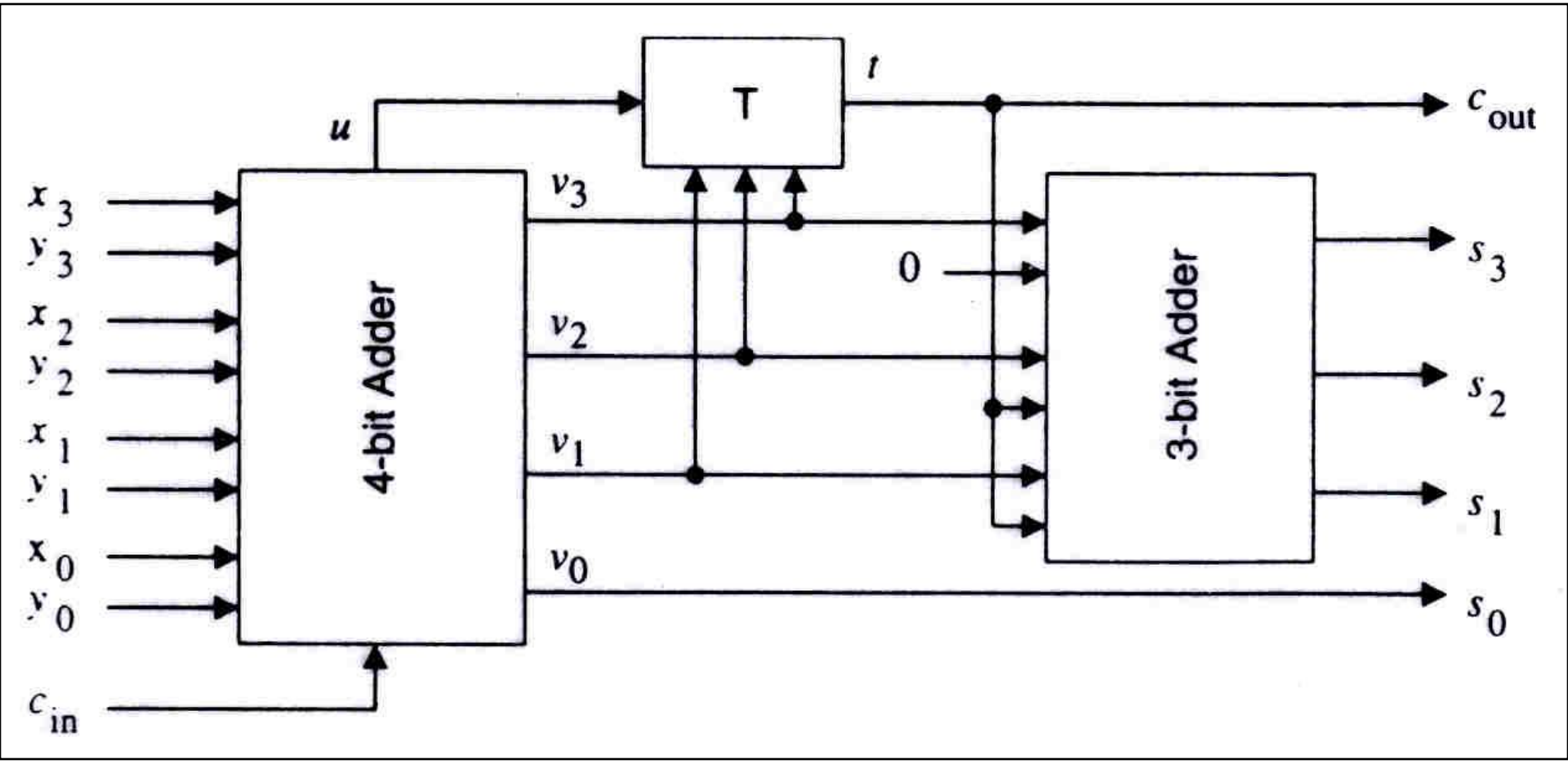
= 0 otherwise

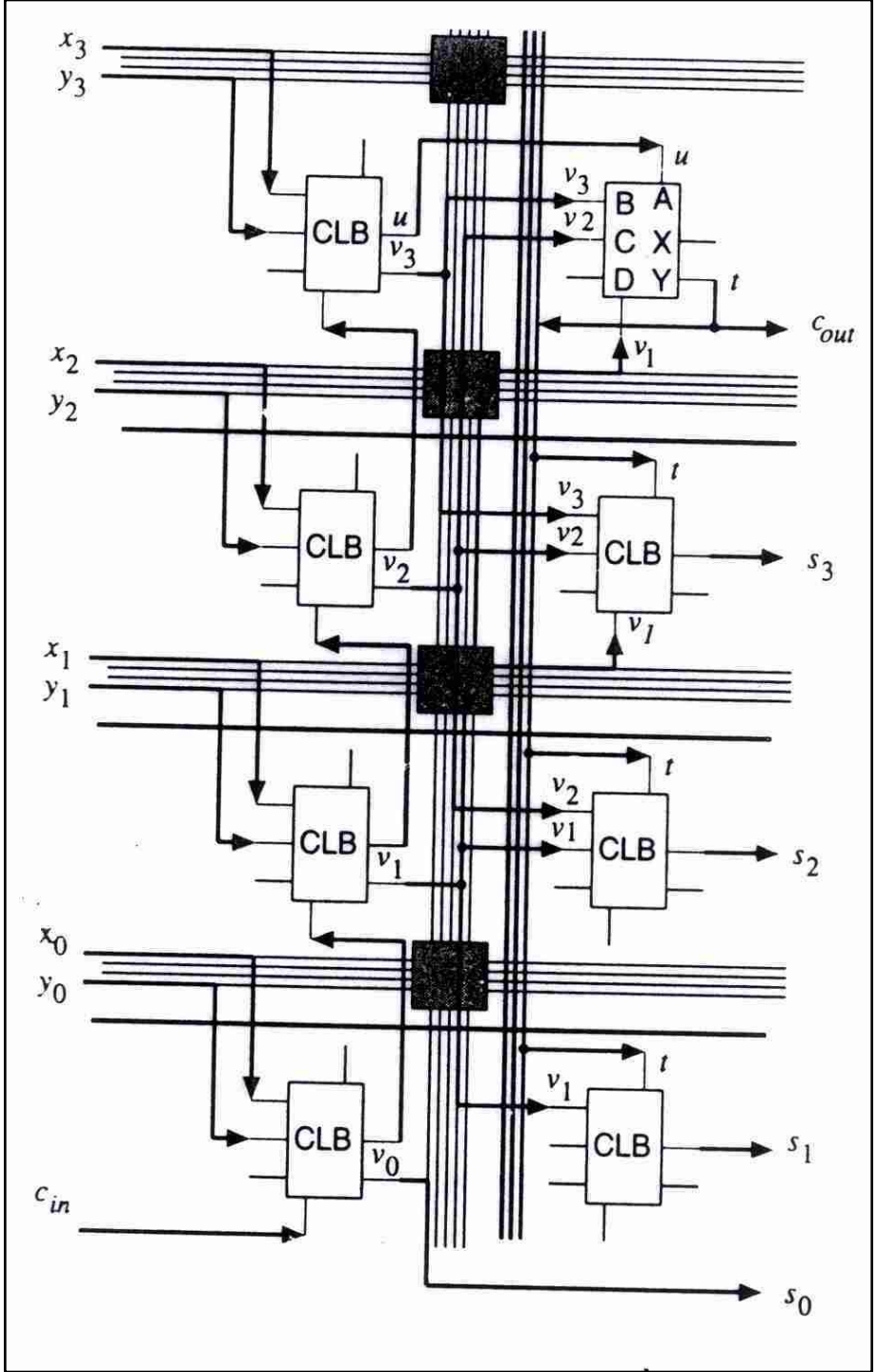
The condition u = 1 or v > = 10 corresponds to the switching function

$t = u + v_3v_2 + v_3v_1$

$s_3 = v_3 \text{ EXOR } t\,(v_2 + v_1)$
$s_2 = v_2 \text{ EXOR } tv_1'$
$s_1 = v_1 \text{ EXOR } t$
$s_0 = v_0$
$c_{out} = t$

- One-Digit BCD Adder




x_3
y_3
CLB
u
v_3
v_3
v_2
B A
C X
D Y
u
t
c_{out}
v_1
x_2
y_2
CLB
v_2
t
v_3
v_2
CLB
s_3
v_1
x_1
y_1
CLB
v_1
t
v_2
v_1
CLB
s_2
x_0
y_0
CLB
v_0
t
v_1
CLB
s_1
c_{in}
s_0

Basic Design steps

Design Entry- Schematic entry or behavioral description using HDL.

Implementation- Partitioning the design into sub modules that can mapped into CLBs, placement of sub modules onto chips, and routing of signals to connect the sub modules.

Design Verification : which uses in circuit testing, simulation and timing analysis.

Behavioral Description

Netlist

Map to FPGA blocks

Bit Stream

Download to FPGAs

FPGA Programming

- VHDL Source Code

```
entity leddcd is
    port(
        d: in std_logic_vector(3 downto 0);
        s: out std_logic_vector(6 downto 0);
    );
end;

architecture leddcd_arch of leddcd is
begin
    s <= "1110111" when d="0000" else
         "0010010" when d="0001" else
         "1101101";
end leddcd_arch;
```