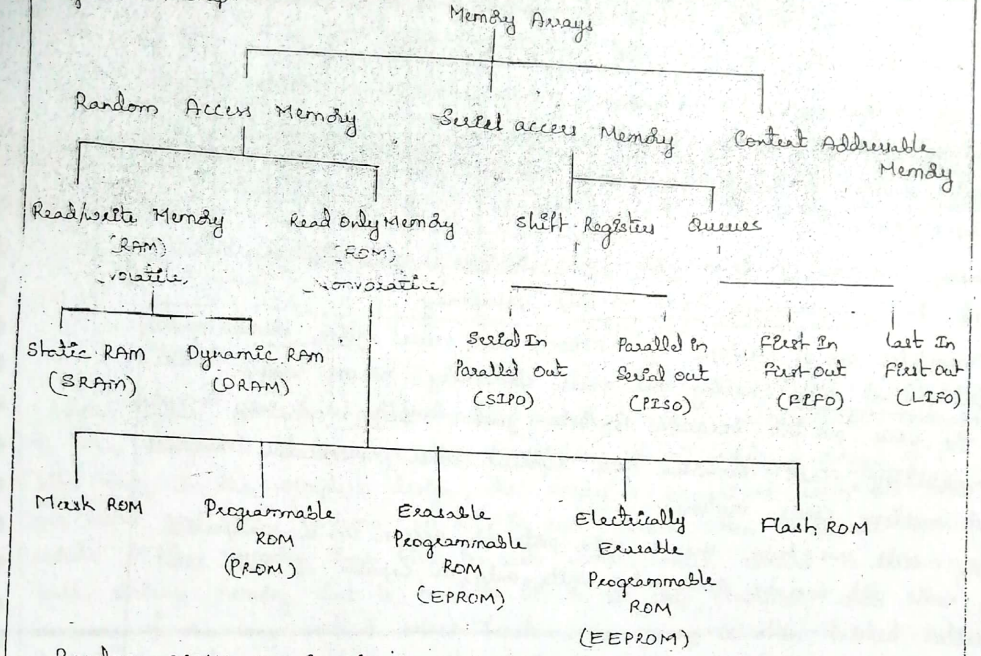
UNIT-6 Sneha 9th 7 B. Mahati ECE-A 460

UNIT-VI Array Subsystem — S. Prashanthi

Introduction :-

Memory arrays often account for the majority of transistors in a CMOS system-on-chip.



Random access memory is accessed with an address and has a latency independent of the address.

Serial access memories are accessed sequentially so no address is necessary.

Content addressable memories determine which address(es) contain data that matches a specified key.

Random access memory is commonly classified as read-only memory (ROM) and read/write memory (RAM). A more useful classification is volatile and nonvolatile memory. Volatile memory retains its data as long as power is applied, while nonvolatile memory will hold data indefinitely. RAM is synonymous with volatile memory while ROM is synonymous with non-volatile memory.

The memory cells used in volatile memories can further be divided into static structures and dynamic structures.

Static cells use some form of feedback to maintain their state, while dynamic cells use charge stored on a floating capacitor through an access transistor. Charge will leak away through the access transistor even while the transistor is OFF, so dynamic cells must be periodically read and rewritten to refresh their state. Static RAMs are faster and less troublesome, but require more area per bit than their dynamic counterparts (DRAMs).

Some nonvolatile memories are indeed read-only. The contents of a mask ROM are hardwired during fabrication and cannot be changed.

A programmable ROM (PROM) can be programmed once after fabrication by blowing on-chip fuses with a special high programming voltage.

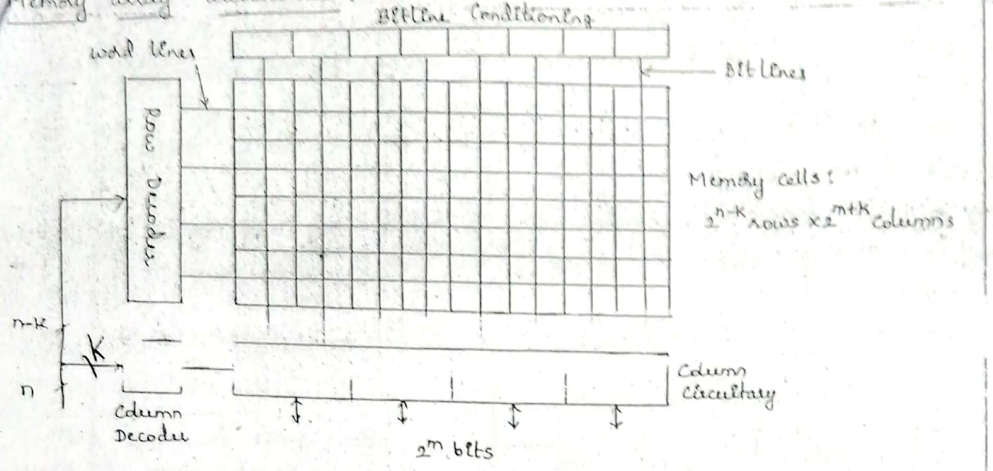
An erasable programmable ROM (EPROM) is programmed by storing charge on a floating gate. It can be erased by exposure to ultraviolet (UV) light for several minutes to knock the charge off the gate. Then the EPROM can be reprogrammed.

Electrically erasable programmable ROMs (EEPROMs) are similar, but can be erased in microseconds with on-chip circuitry.

Flash memories are a variant of EEPROM that erase entire blocks rather than individual bits. Sharing the erase circuitry across larger blocks reduces the area per bit. Because of their good density and easy in-system reprogrammability, Flash memories have replaced other nonvolatile memories in most modern CMOS systems.

Memory cells can have one or more ports for access. On a read/write memory, each port can be read-only, write-only, or capable of both read and write.

Memory array architecture:-



Above figure shows typical small memory array architecture. central to the design is a memory array consisting of 2^n words of storage of 2^m bits each. In the simplest design, the array is organized with one row per word and one column per bit in each word. often there are far more words in the memory than bits in each word. which would lead to a very tall, skinny memory that is hard to fit in the chip floorplan and slow because of the long vertical wires. Therefore, the array is often folded into fewer rows of more columns. After folding, each row of the memory contains 2^k words, so the array is physically organized as 2^{n-k} rows of 2^{m+k} columns of bits. The row decoder activates one of the rows by asserting one of the wordlines. During a read operation, the cells on this wordline drive the bitlines, which may have been conditioned to a known value in advance of the memory access. The column decoder controls a multiplexer in the column circuitry to select 2^m bits from the row as the data to access.

Larger memories are generally built from multiple smaller subarrays so that the wordlines and bitlines remain reasonably short, fast, and low in power dissipation.

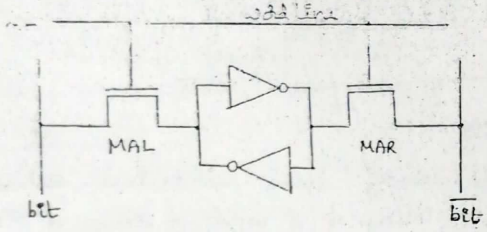
Static RAM :-

SRAM cells use a simple bistable circuit to hold a data bit. A static RAM cell can hold the stored data bit so long as the power is applied to the circuit. SRAMs have three operational modes.

When the cell is in a hold state, the value of the bit is stored in the cell for future usage.

During a write operation, a logic 0 or 1 is fed to the cell for storage.

During a read operation, the value of the stored bit is transmitted to the outside world.



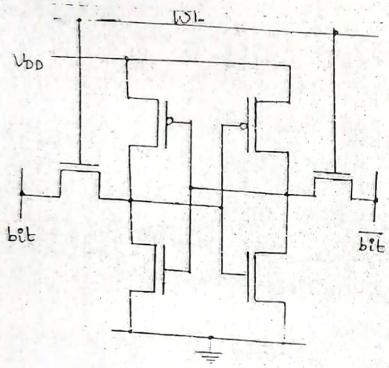
General SRAM cell

Above figure shows basic SRAM cell. A pair of cross-coupled inverters provides the storage, while two access transistors MAL and MAR provide read and write operations. The access transistors are controlled by the word line signal WL that defines the operational modes.

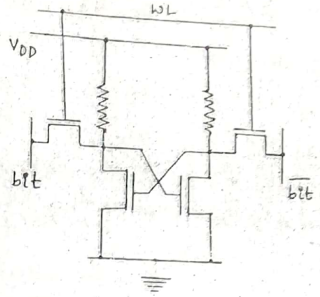
When WL=0, both access FETs are off and the cell is isolated. This defines the hold condition. To perform a read or write operation, the word line is brought up to a value of WL=1. This turns on the access transistors connecting the dual-rail data lines bit and $\overline{bit}$ to the outside circuitry. These are often called the bit and bit-bar lines, respectively.

A write operation is performed by placing voltages on the bit and $\overline{bit}$ lines, which then act as inputs. Dual rail logic helps increase the writing speed.

For a read operation, the bit and bit-bar lines act as outputs and are fed into a sense amplifier that determines the stored value or stored state. The distinction between read and write operations is obtained by circuitry outside the cell array.



6T cell

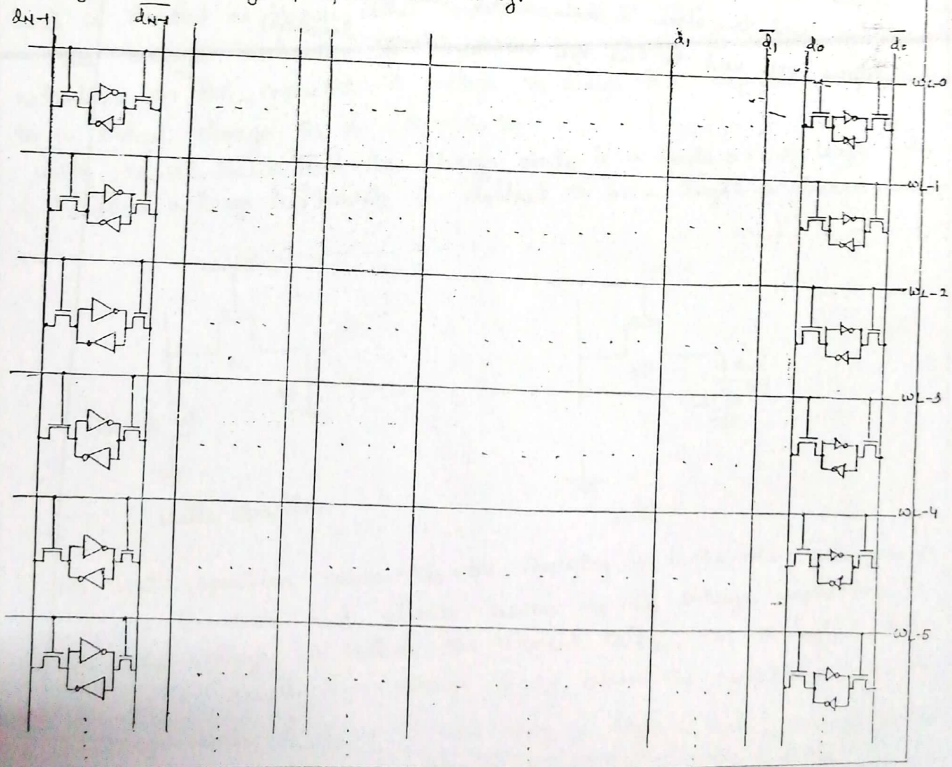


4T cell with poly resistors

Two types of CMOS cells are dominant in practice. 6T cell uses standard CMOS inverters. 4T cell uses resistors as load devices in an nMOS circuit.

SRAM Arrays :-

Static SRAM arrays are created by replicating the basic storage cell and adding the necessary peripheral circuitary.



word lines are assumed to run in a horizontal
and bit-bar lines are patterned vertically.

The outputs of a row decoder provide the word line signals to the storage cells. The address word specifies a particular row, which is then driven high. The access transistors of the selected row cells are turned on, permitting the read/write operations to take place.

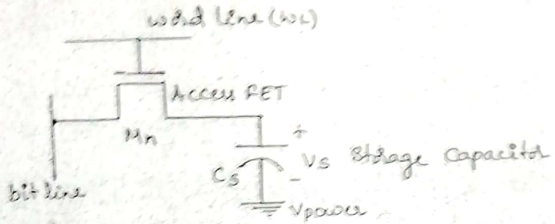
The row decoder outputs are fed into row driver circuits that are used to drive the word lines of the arrays. Drivers are needed because of the large capacitive load presented by the long interconnects and the access transistors connected to each word line.

The input/output bit and bit-bar data lines of the cell for the columns of the memory matrix. The data flow is thus visualized to be vertical for both read and write operations. Once a word line is driven high by the row decoder, every cell in the row is accessible. To choose a particular k-bit word in the row, we must add the group of column decoder circuits that select a particular set of k columns in the matrix.

A read operation requires an output from the cells, so the column decoder circuit acts as multiplexer. For a write operation, the demux mode must be used to steer a data word into the proper columns.

Column drivers are used to feed the MUXes (column decoders).

Dynamic RAM :



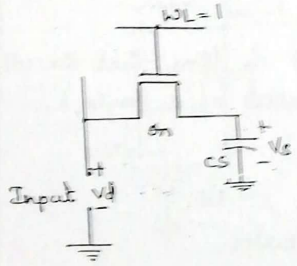
1T DRAM cell

Dynamic RAM (DRAM) cells are substantially smaller than SRAM cells which leads to higher density storage arrays. The reduced cost per bit makes them attractive for applications requiring large read-write memory sizes. DRAMs are slower than SRAMs, and require more peripheral circuitry. At the circuit level, they are simple in structure.

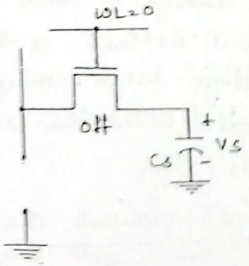
Above figure shows 1T DRAM cell. It consists of a single access nFET Mn and a storage capacitor Cs. The cell is controlled by the word line signal WL and a single bit line provides the I/O path to the cell. The bottom of the capacitor is connected to one of the power supply rails, and is denoted as Vpower, either V_{DD} or V_{SS} may be used.

The storage mechanism is based on the concept of temporary charge retention on the capacitor. A voltage Vs across the capacitor corresponds to a stored charge Q_S of $Q_S = C_S V_S$.

With $V_S = 0V$, $Q_S = 0$ and the charge state is a logic 0. A large value of Vs gives a large Q_S, which is defined to be a logic 1 charge state.



Write Operation



Hold

For write operation $V_{power} = V_{SS} = 0V$. Applying V_{DD} to the nFET gate turns on the access transistor and allows access to the storage capacitor. The input data voltage Vd controls the current to/from Cs. A logic 0 data voltage $V_d = 0V$ results in a voltage $V_S = 0V$ across the capacitor, corresponding to a charge state of $Q_S = 0$. If we apply a logic 1 data voltage $V_d = V_{DD}$

equal to the power supply, the voltage on the gate reduces the transmitted signal by an nFET threshold voltage. The largest voltage that can be passed to the capacitor is

$$V_S = V_{max} = V_{DD} - V_{Tn}$$

which gives a maximum charge of $Q_{max} = C_S(V_{DD} - V_{Tn})$

The hold state is achieved by turning off the access transistor with a word line signal of $WL = 0$.

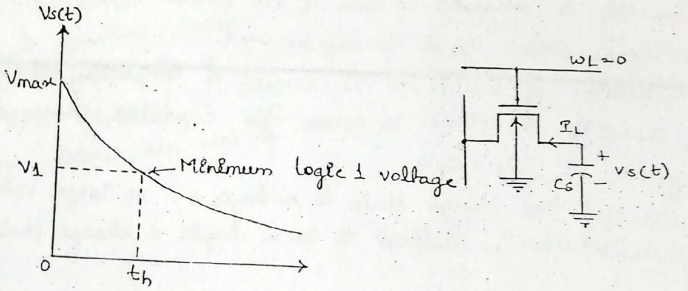
The dynamic aspect of the cell arises during a data hold time. A MOSFET that is biased into cutoff with $V_G < V_T$ still admits small leakage currents.

A logic 1 voltage $V_S = V_{max}$ on the storage capacitor provides the electromotive force for the leakage current I_L flowing away from C_S. This can be described by

$$I_L = -\left(\frac{dQ_S}{dt}\right)$$

which shows that the current removes charge from the capacitor.

$$\therefore I_L = -C_S\left(\frac{dV_S}{dt}\right)$$ so that V_S also drops.



Charge leakage in DRAM cell.

The hold time t_h is defined as the longest period of time that the cell can maintain a voltage large enough to be interpreted as a logic 1, the hold time is also called the retention time.

$$I_L = -C_S\left(\frac{\Delta V_S}{\Delta t}\right)$$

where ΔV_S and Δt represent changes in the variables.

$$\boxed{t_h = |\Delta t| = \left(\frac{C_S}{I_L}\right)\Delta V_S}$$

Above expression shows that the hold time may be increased by using a large capacitance and minimizing the leakage current.

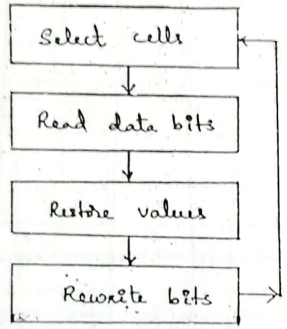
Refresh Operation

Memory units must be able to hold data so long as the power is applied. To overcome the charge leakage problem, DRAM arrays employ a refresh operation where the data is periodically read from every cell, amplified, and then rewritten.

The cycle must be performed on every cell in the array with a minimum refresh frequency of about

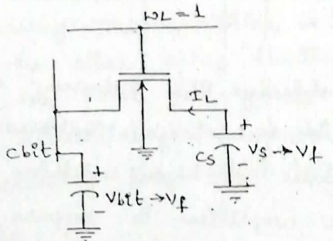
$$f_{refresh} = \frac{1}{2t_h}$$

Refresh circuitry is included in the overhead logic that surrounds the cell array. The refresh cycle is designed to operate in the background and is therefore transparent to the user.



Refresh operation

Read Operation in a DRAM cell :-



The voltage Vs on the capacitor at the read time provides the voltage to move charge from Cs to the bit line capacitance Cbit, which sets up a charge sharing situation. Cbit includes the line capacitance and other parasitic contributions such as the input capacitance of the sense amplifier.

The initial charge on the capacitor is

$$Q_s = C_s V_s$$

where $V_s = 0V$ for a logic 0, and $V_s > 0$ for a logic 1.

Current flows from C_s to C_{bit} continues until the voltages are equal to the final voltage V_f. $V_{bit} = V_s$. The charge is redistributed according to

$$Q_s = C_s V_f + C_{bit} V_f$$

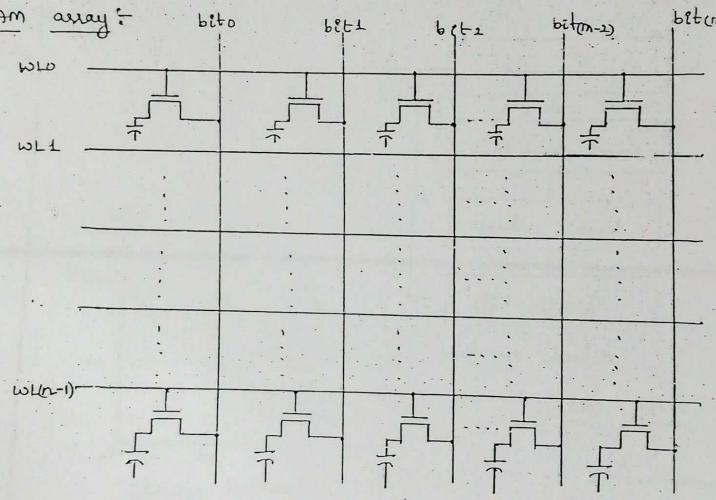
The initial and final values of Q_s must be equal by charge conservation, so

$$V_f = \frac{Q_s}{C_s + C_{bit}}$$

$$\therefore \boxed{V_f = \left[\frac{C_s}{C_s + C_{bit}}\right] V_s}$$

It shows that $V_f < V_s$ for a stored logic 1. In practice, V_f is usually reduced to a few tenths of a volt, so that the noise margin of the sense amplifier becomes a critical factor.

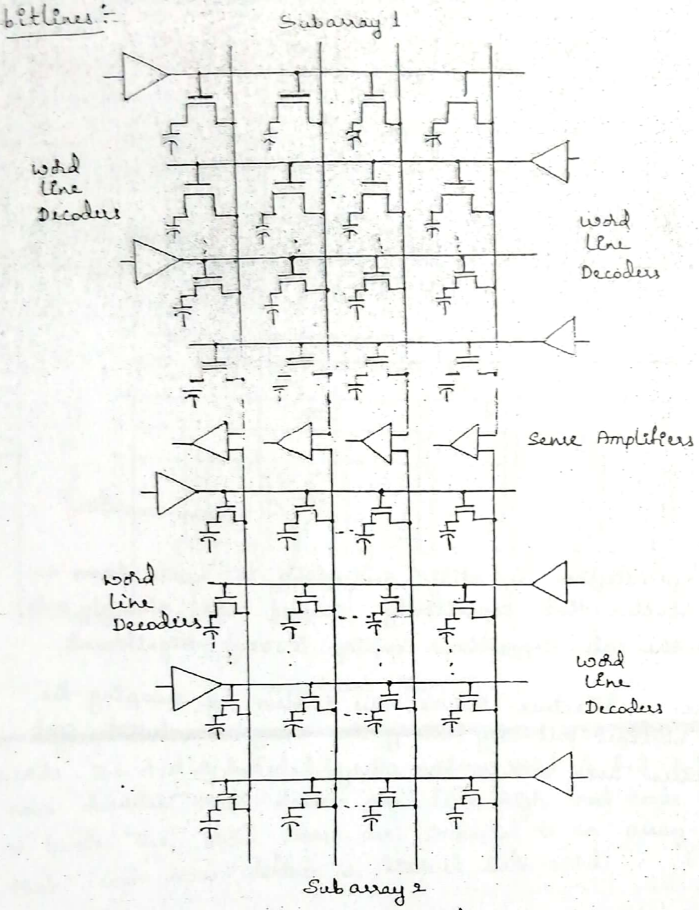
DRAM array :-



Large DRAMs are divided into multiple subarrays. The subarray size represents a tradeoff between density and performance. Larger subarrays are slow and have small bitline swings because of high wordline and bitline capacitance. The array uses a sense amplifier to compare the bitline voltage to that of an idle bit line (precharged to $V_{DD}/2$). The sense amplifier must also be very compact to fit the tight pitch of the array. The low-swing bitlines are very sensitive to noise.

These bitline architectures, open, folded and twisted, offer different compromises between noise and area.

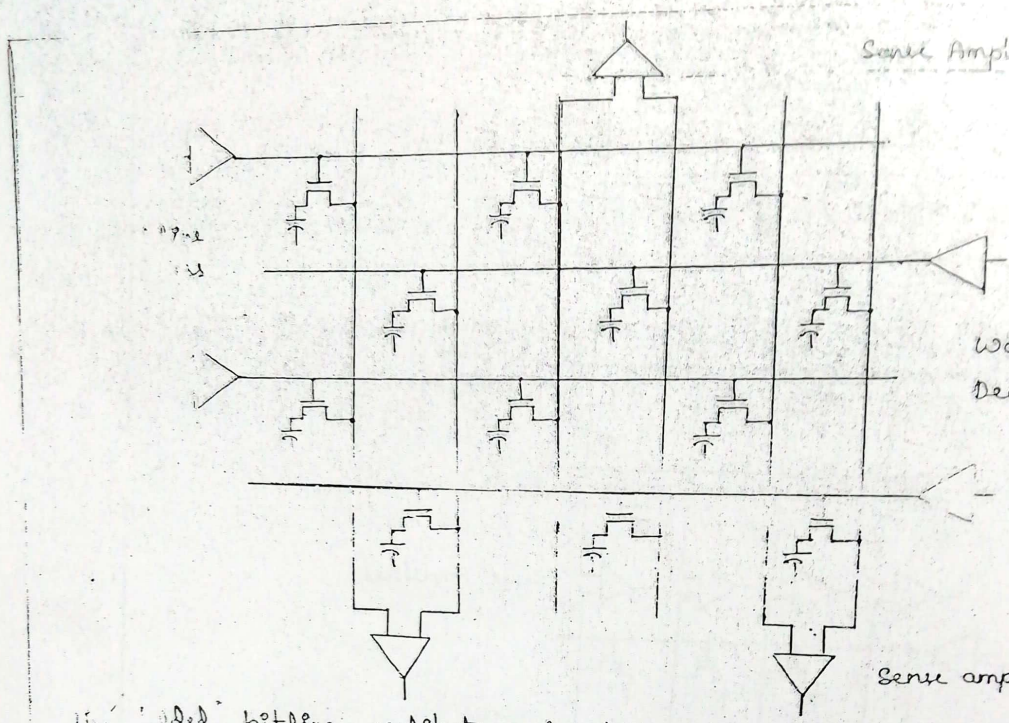
Open bitlines :-



In this architecture, the sense amplifier receives one bitline from each of two subarrays. The wordline is only asserted in one array, leaving the bitlines in the other array floating at the reference voltage. The arrays are very dense. Any noise that effects one array more than the other will appear as differential noise at the sense amplifier. Thus, open bit lines have unacceptably low signal-to-noise ratios for high-density DRAM.

Folded bitlines :-

In this architecture, each bitline connects to only half as many cells. Adjacent bitlines are organized in pairs as inputs to the sense amplifiers. When a wordline is asserted, one bitline will switch while its neighbor serves as the quiet reference. Many noise sources will couple equally onto the two adjacent bitlines so they tend to appear as common mode noise that is rejected by the sense amplifier. The noise advantage comes at the expense of greater layout area.

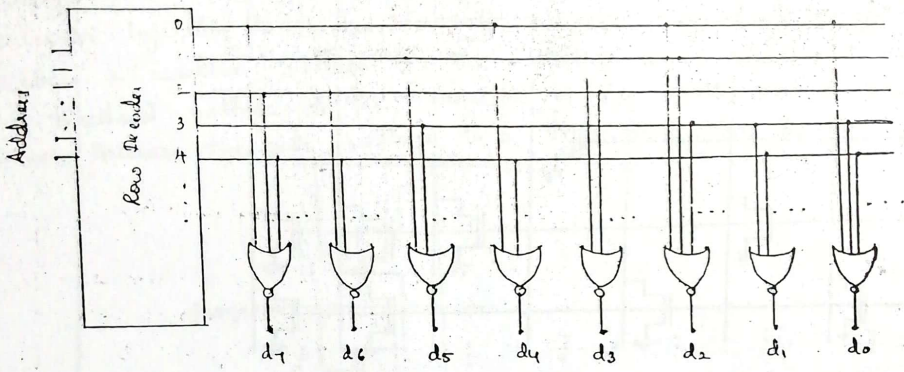


...ded bitline architecture is still susceptible to noise from a ...ng switching bitline that capacitively couples more strongly onto ... bitlines in the pair. Capacitive coupling is very significant ... processes.

...sted bitline architecture solves this problem by swapping the ... the folded bitlines part way along the array. The twisted cost ... amount of extra area within the array.

ROM arrays:-

Read only memories are used for permanent bit storage. The structure of a ROM array is similar to that used for RAMs, but the individual bit cells are much simpler. The data stored in a basic read only memory is created by the selective placement of FETs. Since this is accomplished in the physical design, the data cannot be altered once the chip is fabricated.



NOR based ROM

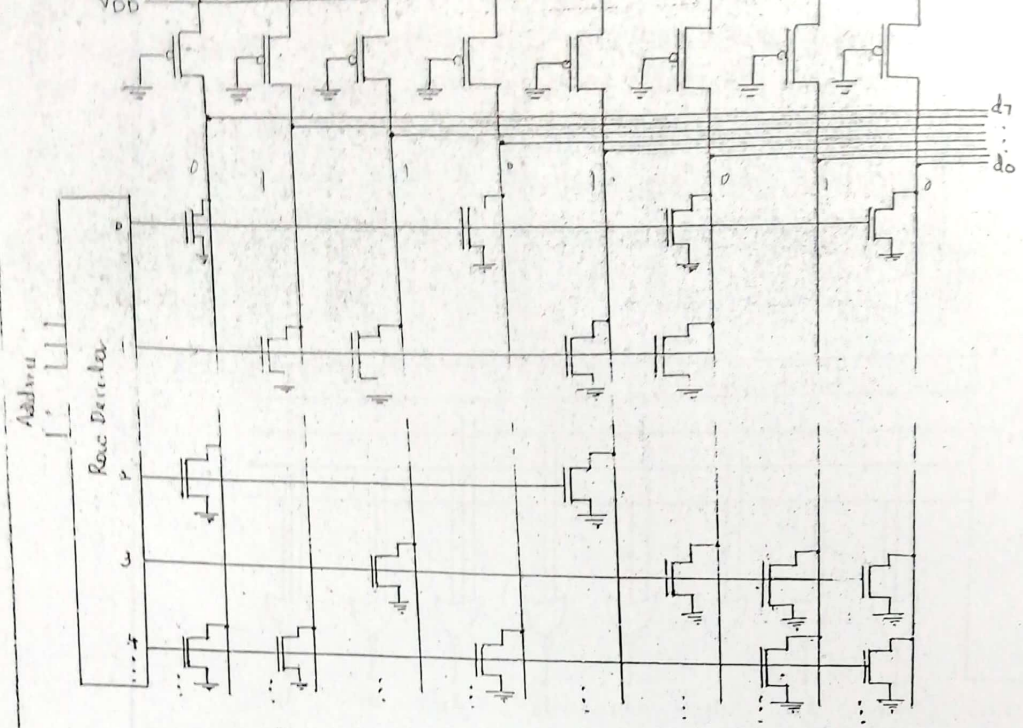
Above figure shows a ROM array that uses NOR gates to store 8-bit data words $D = d_7d_6d_5d_4d_3d_2d_1d_0$. An address word is fed into an active high row decoder that drives one line high and keeps the others at logic 0 levels. The word lines are connected to an array of NOR gates such that each row defines a distinct data word.

Ex:-

Row	Data
0	01101010
1	10010011
2	01110111
3	11011000
4	00101100

ROM array using pseudo-nMOS:-

Pseudo-nMOS uses only one pull-up pFET for each NOR gate, the task of programming centers on placement of nFETs that act as pull down devices. A logic 0 output is obtained by providing a FET with its gate connected to the driving word line. When a pull-down transistor turns on, it provides a good connection to ground and pulls the output low.

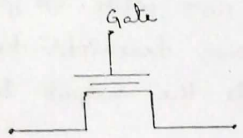


ROM array using pseudo-nMOS circuitry (NOR)

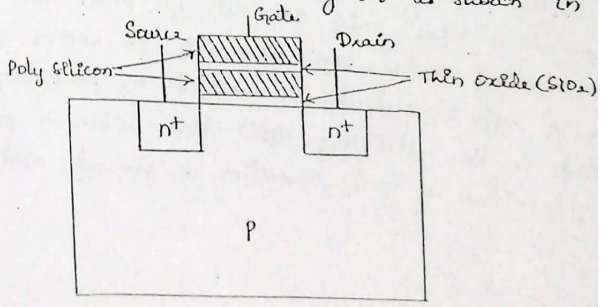
User programmable ROMs :-

Electrically programmable ROMs (EPROMs) allow the user to store data as required by the application. Special voltage settings are used to write to the cells. Read operations are performed with nominal voltage levels, so that the data remains unchanged. Many ROM devices provide for erasure and resetting the contents of the array. Optical erasure using UV light was used in early EPROM designs, but these have been replaced by electrically alterable devices.

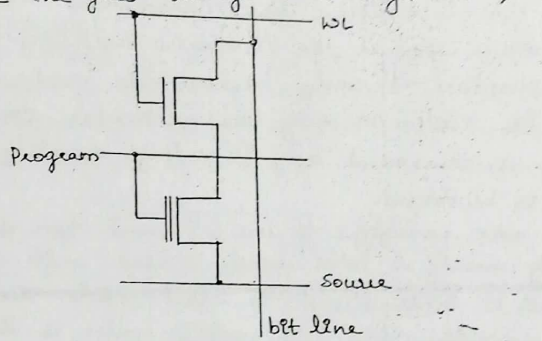
A reprogrammable ROM array is built using special FETs that use a pair of stacked poly gates and has the circuit symbol as shown in figure.



Electrically Floating Gate

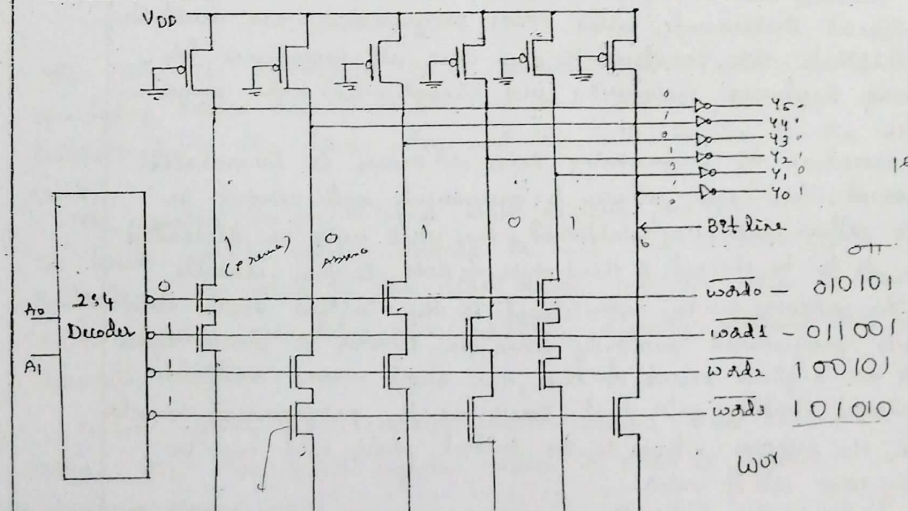


The topmost gate constitutes the usual gate terminal of the transistor. Another polygate layer is sandwiched in between the top poly and the silicon substrate. It is not electrically connected to any part of the transistor, and is therefore called an electrically floating gate. The floating gate is used to store negative electron charge, which increases the threshold voltage of the transistor above the normal value. Applying a high voltage to the upper gate causes electrons to jump through the thin oxide onto the floating gate through the process called avalanche injection or Fowler-Nordheim emission. Injecting the electrons induces a negative voltage on the floating gate, effectively increasing the threshold voltage of the transistor to the point that is always OFF. If we increase the gate voltage to a high value, then the FET goes active.



EEPROM cell with write line

Pseudo-nMOS NAND ROM:-



NAND ROM uses active low wordlines. Transistors are placed in series and the transistors on the nonselected rows are ON. If the no transistor is associated with the selected word, the bitline will pull down. If a transistor is present, the bit line will remain high.

In NOR ROM the size of the cell is limited by the ground line. In NAND ROM delay grows gradually with the number of series transistors discharging the bitline.

Types of ROMs :-

The required paths in a ROM may be programmed in four different ways. First is called mask programming and is done by the semiconductor company during the last fabrication process of the unit. The procedure for fabricating a ROM requires that the customer fill out the truth table he wishes the ROM to satisfy. The manufacturer makes the corresponding mask for the paths to produce the 1's and 0's according to the customer's truth table. This procedure is costly because the vendor charges the customer a special fee for custom masking the particular ROM. For this reason, mask programming is economical only if a large quantity of the same ROM configuration is to be ordered.

For small quantities, it is more economical to use a second type of ROM called programmable read-only memory or PROM. When ordered units contain all the fuses intact, giving all 1's in the bits of the stored words. The fuses in the PROM are blown by application of a high-voltage pulse to the device through a special pin. A blown fuse defines a binary 0 state and an intact fuse gives a binary 1 state. This allows the user to program the PROM in the laboratory to achieve the desired relationship between input addresses and stored words. Special instruments called PROM programmers are available commercially to facilitate this procedure. In any case, all procedures for programming ROMs are hardware procedures even though the word programming is used.

The hardware procedure for programming ROMs or PROMs is irreversible and, once programmed, the fixed pattern is permanent and cannot be altered. Once a bit pattern has been established, the unit must be discarded if the bit pattern is to be changed. A third type of ROM is the erasable PROM or EPROM. The EPROM can be restructured to the initial state even though it has been programmed previously. When the EPROM is placed under a special UV light for a given period of time, the short wave radiation discharges the internal floating gates that serve as the programmed connections. After erasure, the EPROM returns to its initial state and can be reprogrammed to a new set of values.

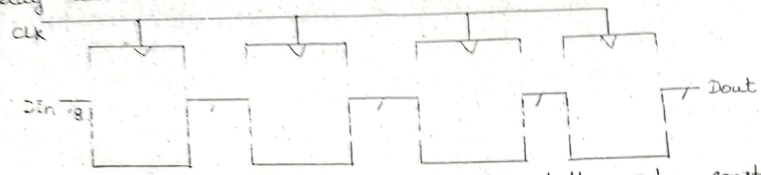
The fourth type of ROM is the electrically erasable PROM (EEPROM). It is like the EPROM except that the previously programmed connections can be erased with an electrical signal instead of UV light. The advantage is that the device can be erased without removing it from its socket.

Serial Access Memories:-

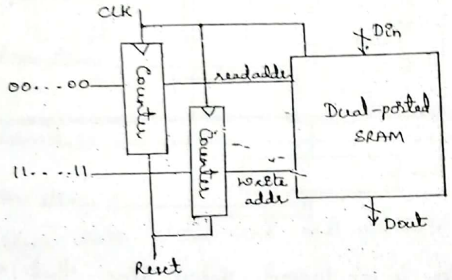
Using the basic SRAM cell and/or registers, we can construct a variety of serial access memories including shift registers and queues. These memories avoid the need for external logic to track addresses for reading & writing.

Shift Registers:-

A shift register is commonly used in signal processing applications to store and delay data.



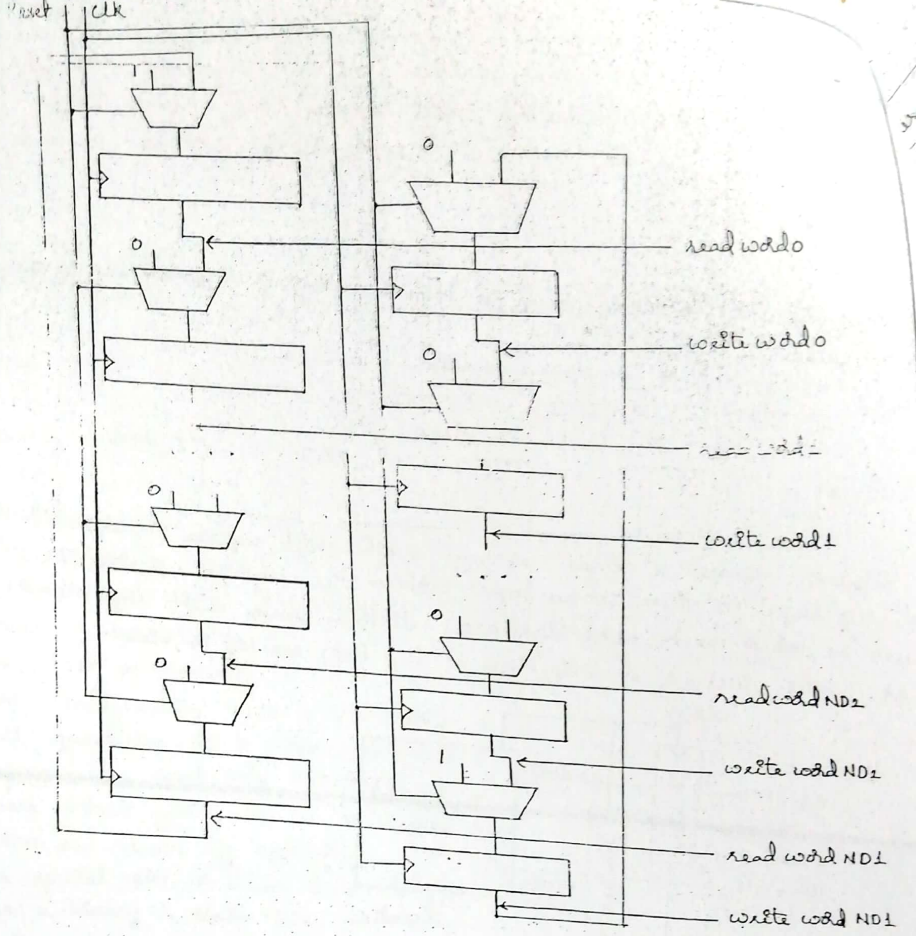
Above figure shows a simple 4-stage, 8-bit shift register constructed from 32 flip flops. As there is no logic between the registers, particular care must be taken that hold times are satisfied. Dense shift registers uses dual port RAMs instead of Flip Flops as flip flops are big in size.



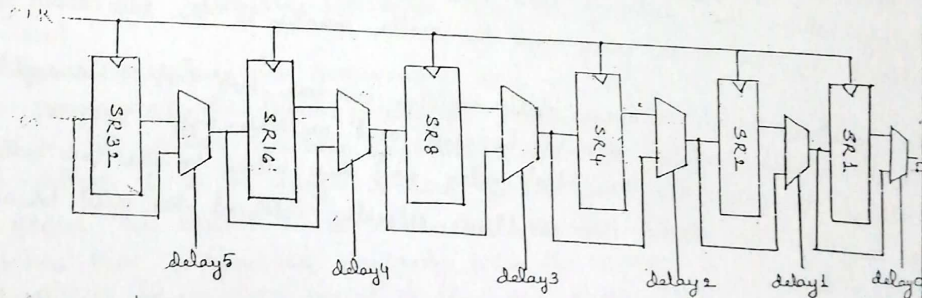
The RAM is configured as a circular buffer with a pair of counters specifying where the data is read and written. The read counter is initialized to the first entry and the write counter to the last entry on reset.

The counters in an N stage shift register can use two 1-of-N hot registers to track which entries should be read and written. Again one is initialized to point to the first entry and the other to the last entry. These registers can drive the wordlines directly without the need for a separate decoder.

N hot register is a register which contains '1' as loaded (in one flip flop) value Ex: For 4 bit register 1000 & 0001 is loaded.

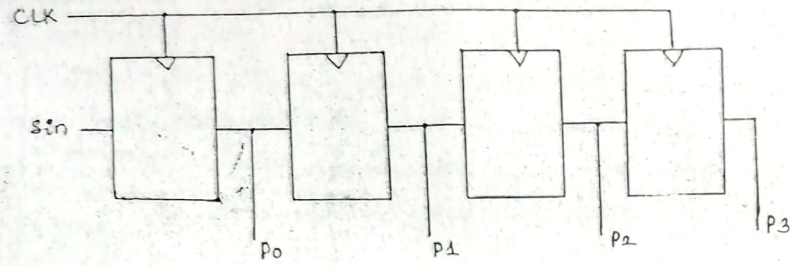


...iant of a shift register is a tapped delay line that offers ... number of stages of delay.

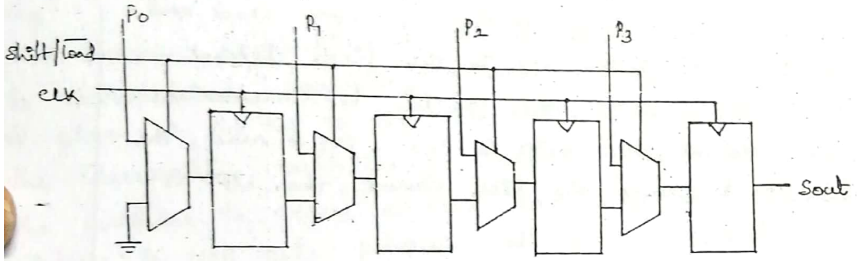


... figure shows a 64 stage tapped delay line that could be used in ... processing system. Delay blocks are built from 32, 16, 8, 4, 2 and 1 stage shift registers. Multiplexers control pass-around of the delay blocks to provide the ... selected total delay.

Another variant is a serial/parallel memory. There are useful in signal processing and communications systems.



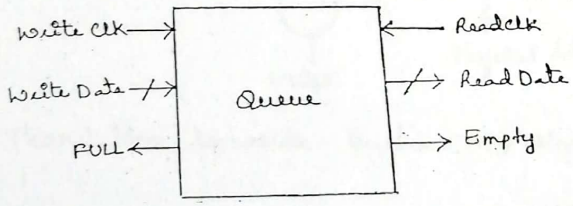
Above figure shows 4-stage serial In parallel out (SIPO) memory



Above figure shows parallel in serial out (PISO) memory. It contains 4 stages.

Queues :-

Queues allow data to be read and written at different rates.



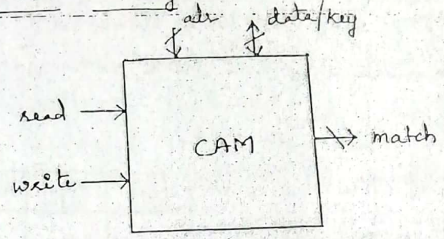
Above figure shows an interface to a queue. The read and write operations each are controlled by their own clocks that may be asynchronous. The queue asserts the FULL flag when there is no room remaining to write data and the EMPTY flag when there is no data to read. Because of other system delays, some queues also provide ALMOST-FULL and ALMOST-EMPTY flags to communicate the impending state and halt write or read requests. The queue internally maintains read and write pointers indicating which data should be accessed next.

The pointers can be counters or 1-of-N hot registers.

First In First Out (FIFO) queues are commonly used to buffer data between two asynchronous streams. The FIFO is organized as a circular buffer. On reset, the read and write pointers are both initialized to the first element and the FIFO is EMPTY. On a write, the write pointer advances to the next element. If it is about to catch read pointer, the FIFO is FULL. On a read, the read pointer advances to the next element. If it catches the write pointer, the FIFO is EMPTY again.

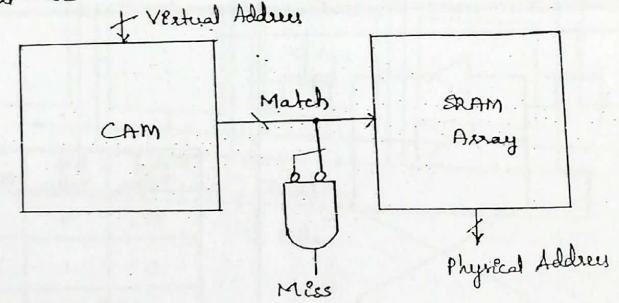
Last In First Out (LIFO) queues also known as stacks are used in applications such as subroutine or interrupt stacks in microcontrollers. The LIFO uses a single pointer for both read and write. On reset, the pointer is initialized to the first element and the LIFO is EMPTY. On a write, the pointer is incremented. If it reaches the last element, the LIFO is FULL. On a read, the pointer is decremented. If it reaches the first element, the LIFO is EMPTY again.

Content-addressable Memory :-



Above figure shows symbol for a content-addressable memory (CAM). The CAM acts as an ordinary SRAM that can read or written given the address and data but also performs matching operations. Matching asserts a matchline output for each word of the CAM that contains a specified key.

A common application of CAM is translation lookaside buffers (TLBs) in microprocessor supporting virtual memory. The virtual address is given as the key to the TLB CAM. If this address is in the CAM, the corresponding matchline is asserted. This matchline can serve as the wordline to access a RAM containing the associated physical address. A NOR gate processing all of the matchlines generates a Miss signal for the CAM.

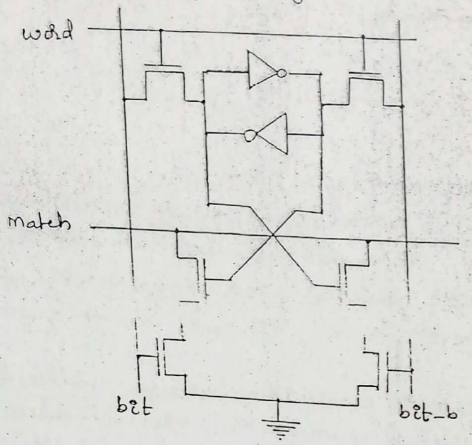


Translation lookaside Buffer (TLB) using CAM

CAM cell :-

10T and 9T CAM cells consists of a normal SRAM cell with additional transistors to perform the match. Multiple CAM cells in the same word are tied to the same matchline. The matchline is either precharged or pulled high as a distributed pseudo-nMOS gate. The key is placed on the bitlines. If the key and the value stored in the cell differ, the matchline will be pulled down. Only if all of the key bits match all of the bits stored in the

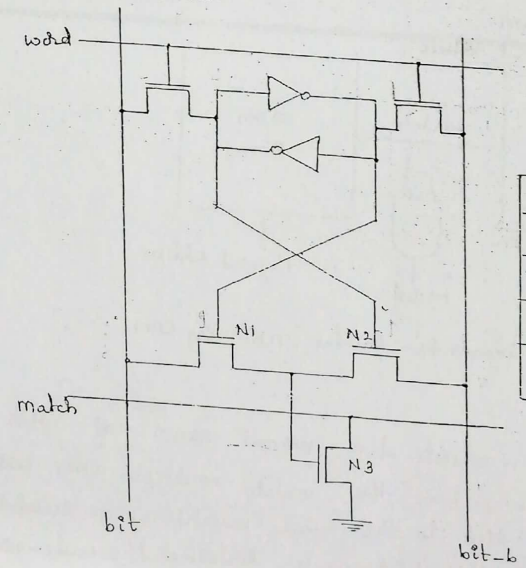
word of memory then the matchline for that word remain high. The key can contain a "don't care" by setting both bit and bit-b low.



Stored value	Bit value	Match
0	0	1
0	1	0
1	0	0
1	1	1

10T CAM cell

CAMs generally have about twice the area of SRAM cells. 9T CAM cell is designed with 9 Transistors. N1 and N2 perform an XOR of the key and cell data. If the values disagree, N3 is turned on to pull down the matchline.

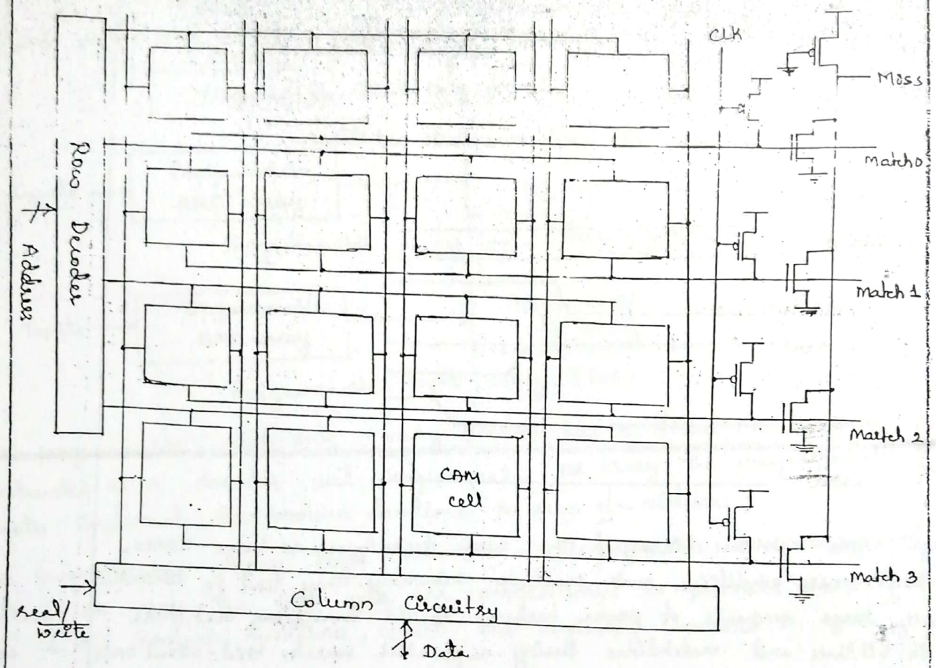


Stored value	Bit value	Match
0	0	1
0	1	0
1	0	0
1	1	1

9T CAM cell

CAM array:

Like an SRAM, CAM array consists of an array of cells, a decoder, and column circuitary. Each row produces a dynamic matchline. The matchlines are precharged with the clocked pMOS transistors. the MISS signal is produced with a distributed pseudo-nMOS NOR.



When the matchlines are used to access a RAM, the monotonicity problem must be considered. Initially, all the matchlines are high. During CAM operation, the lines pull down, leaving at most one line asserted to indicate which row contains the key. The RAM requires a monotonically rising wordline. Strobed AND gates are used to drive the wordlines as early as possible after the matchlines have settled. The strobe can be timed with an inverter chain or replica delay line in the same way that the sense amplifier clock for a SRAM was generated. Clocked sense amplifier consumes power only when activated, but requires a timing chain to activate at the proper time.

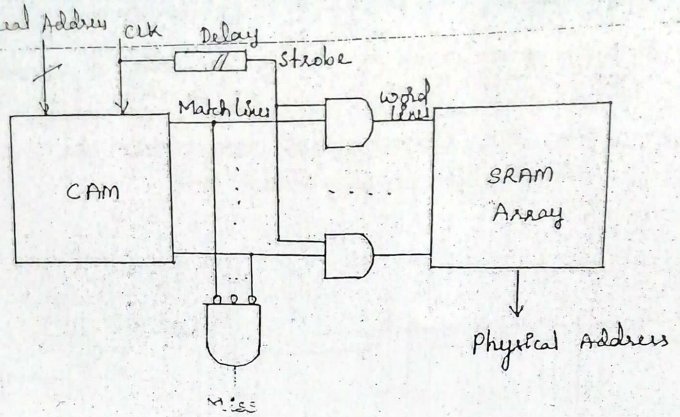
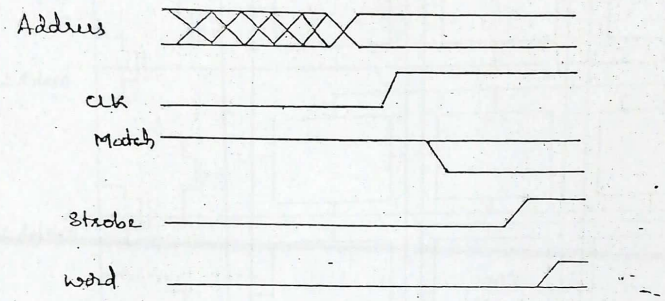


Fig: TLB path with monotonic wordlines

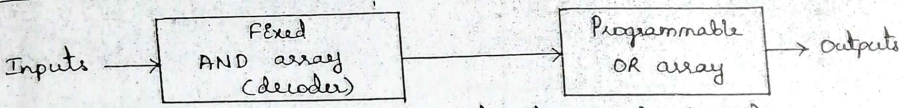


Large CAMs can use many of the same techniques as large RAMs, including sense amplifiers and multiple subarrays. they tend to consume relatively large amounts of power because of the transitions that take place on bitlines and matchlines during a parallel search. Med-sized CAM using a PMOS match-line driver reduces the swing of matchlines and saves the power.

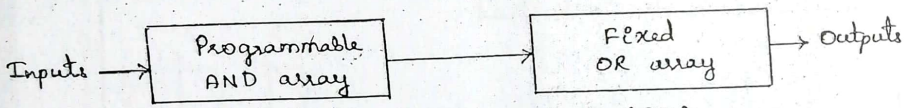
ional Programmable logic Devices :-

mbinational Programmable logic Device (PLD) is an integrated circuit with mable gates divided into an AND array and an OR array to pro- an AND-OR sum of product implementation. There are three major ypes of combinational PLDs and they differ in the placement of the progra- nmable connections in the AND-OR array.

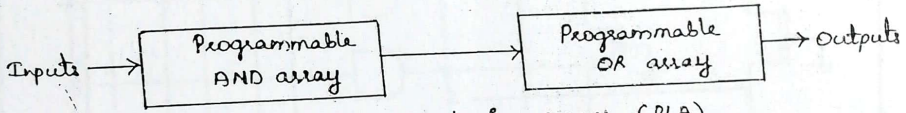
Basic Configuration of three PLDs :-



Programmable Read Only memory (PROM)



Programmable array logic (PAL)



Programmable logic array (PLA)

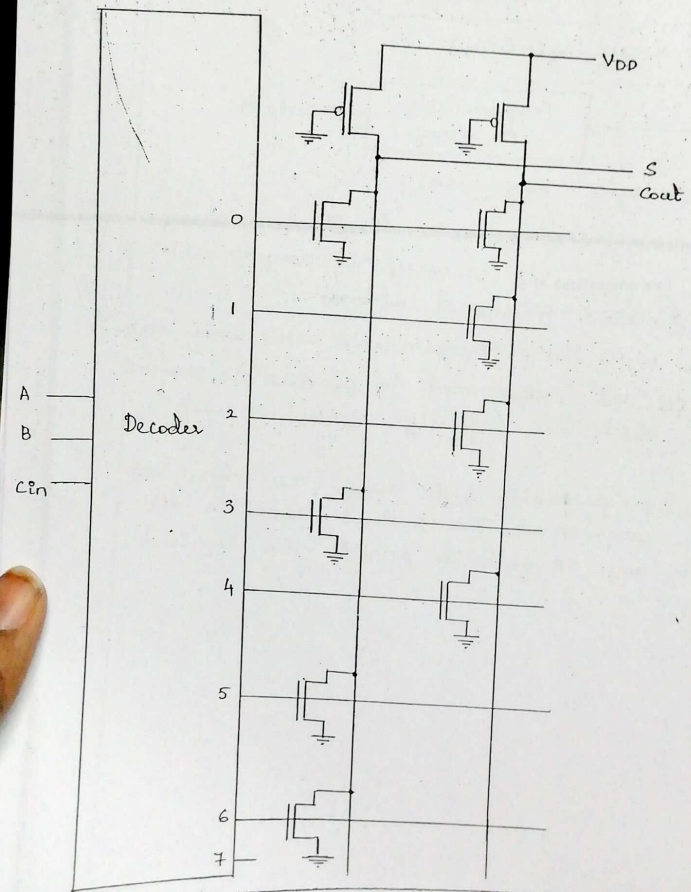
The programmable read-only memory (PROM) has a fixed AND array constructed as a decoder and programmable OR array. The programmable OR gates implement the boolean functions in sum of minterms.

The programmable array logic (PAL) has a programmable AND array and a fixed OR array. The AND gates are programmed to provide the product terms for the boolean functions, which are logically summed in each OR gate.

The most flexible PLD is the programmable logic array (PLA) where both the AND and OR arrays can be programmed. The product terms in the AND array may be shared by any OR gate to provide the required sum of products implementation.

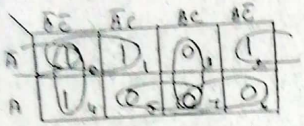
Implementation of Full adder using PROM :-

A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



Implement the following two boolean functions with a PLA:

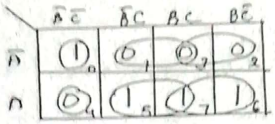
$F_1(A,B,C) = \Sigma m(0,1,2,4)$, $F_2(A,B,C) = \Sigma m(0,5,6,7)$



$F_1 = \bar{A}\bar{C} + \bar{A}\bar{B} + \bar{B}\bar{C}$

$F_1' = AC + AB + BC$

$\therefore F_1 = \overline{AC + AB + BC}$



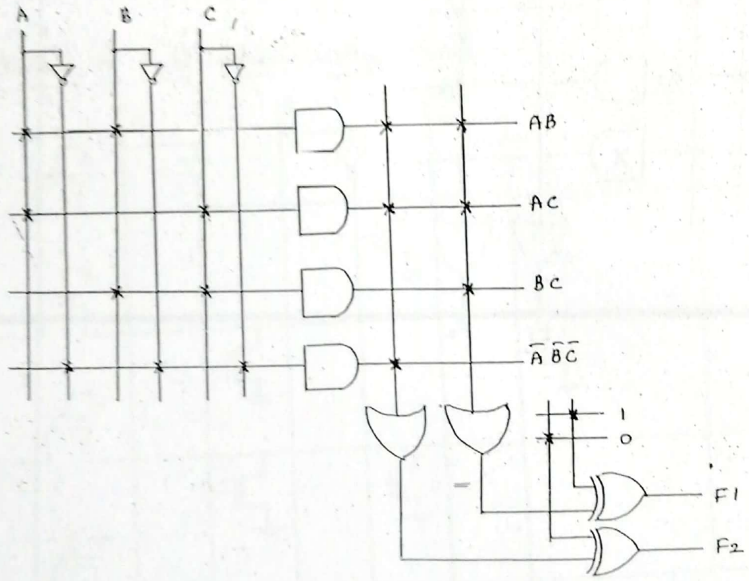
$F_2 = \bar{A}\bar{B}\bar{C} + AC + AB$

$F_2' = \bar{A}C + \bar{A}B + A\bar{B}\bar{C}$

$\therefore F_2 = \overline{\bar{A}C + \bar{A}B + A\bar{B}\bar{C}}$

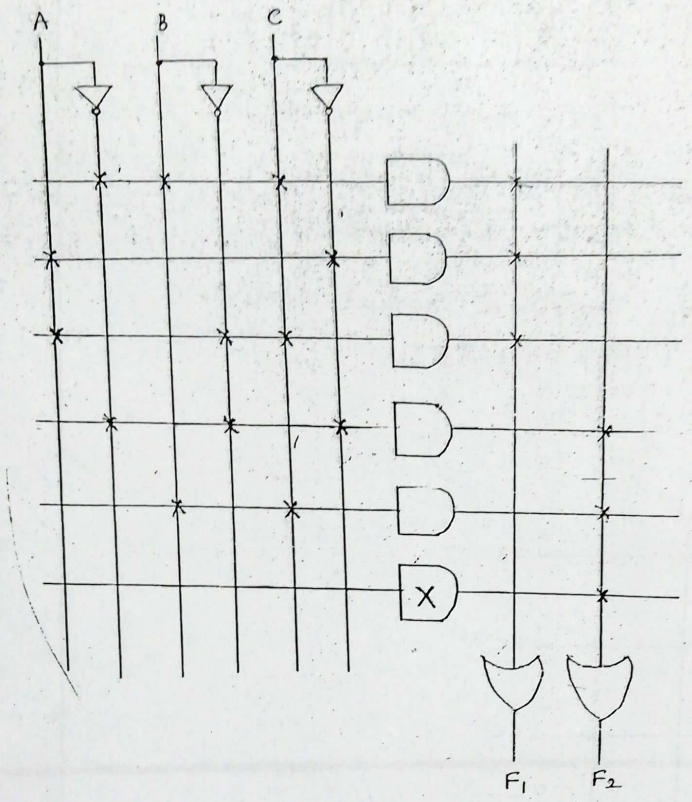
The combination that gives minimum number of product terms is

$F_1 = \overline{AC + AB + BC}$ and $F_2 = AB + AC + \bar{A}\bar{B}\bar{C}$



1 Implement following functions with PAL:

$F_1 = \bar{A}BC + A\bar{C} + A\bar{B}C$ and $F_2 = \bar{A}\,\bar{B}\bar{C} + BC$



A

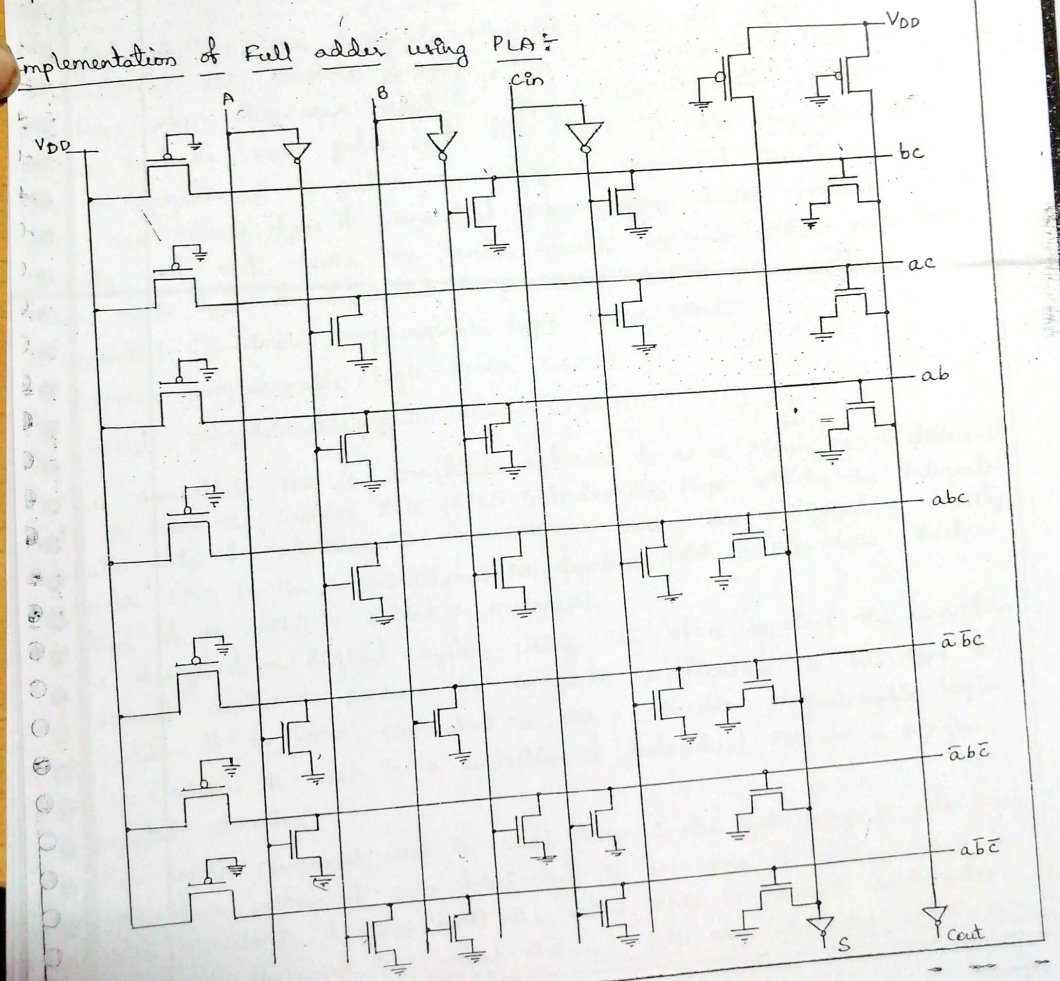
B

able logic array:-

rogrammable logic array (PLA) is similar to the PROM in concept but the PLA does not provide full decoding of the variables does not generate all the minterms. The decoder is replaced by an ray of AND gates that can be programmed to generate any product ns of the input variables. The product terms are then connected to r gates to provide the sum of products for the required Boolean functions.

The most straightforward PLA uses a pseudo-nMOS NOR gate. Advantages of this PLA include simplicity and small size. Disadvantages include the static power dissipation of the NOR gates and slow pull-up response.

Implementation of Full adder using PLA:-



when implementing a combinational circuit with a PLA, careful investigation must be undertaken in order to reduce the number of distinct product terms, since a PLA has a finite number of AND gates. This can be done by simplifying each boolean function to a minimum number of terms. The number of literals in a term is not important since all the input variables are available anyway. Both the true and complement of each function should be simplified to see which one can be expressed with fewer product terms and which one provides product terms that are common to other functions.

...ble Array Logic :-

...programmable array logic (PAL) is a programmable logic device with ...ed OR array and a programmable AND array. Because only the AND ... are programmable, the PAL is easier to program, but is not as ...xible as the PLA.

When designing with a PAL, the boolean functions must be simplified to fit into each section. Unlike the PLA, a product term cannot be shared among two or more OR gates. Therefore, each function can be simplified by itself without regard to common product terms. The number of product terms in each section is fixed.

Sequential Programmable devices :-

Digital systems are designed using flip-flops and gates. Since the combinational PLD consists of only gates, it is necessary to include external flip flops when they are used in the design. Sequential programmable devices include both gates and flip flops. In this way, the device can be programmed to perform a variety of sequential circuit functions. There are several types of sequential programmable devices available commercially and each device has vendor specific variant within each type. Three major types of sequential Programmable devices are

1) Sequential (or simple) programmable logic device (SPLD)
2) Complex programmable logic device (CPLD)
3) Field programmable gate array (FPGA)

The sequential PLD is sometimes referred to as a simple PLD to differentiate it from the complex PLD. SPLD includes flip-flops within the integrated circuit chip in addition to the AND-OR array. The configuration mostly used for SPLD is the combinational PAL together with D Flip Flops. Each section of an SPLD is called a macrocell.

The design of a digital system using PLD often requires the connection of several devices to produce the complete specification. For this type of application, it is more economical to use a complex programmable logic device (CPLD). A CPLD is a collection of individual PLDs on a single integrated circuit.

The basic component used in VLSI design is the gate array. A gate array consists of a pattern of gates fabricated in an area of silicon that is repeated thousands of times until the entire chip is covered with gates. Arrays of one thousand to hundred thousand gates are fabricated within a

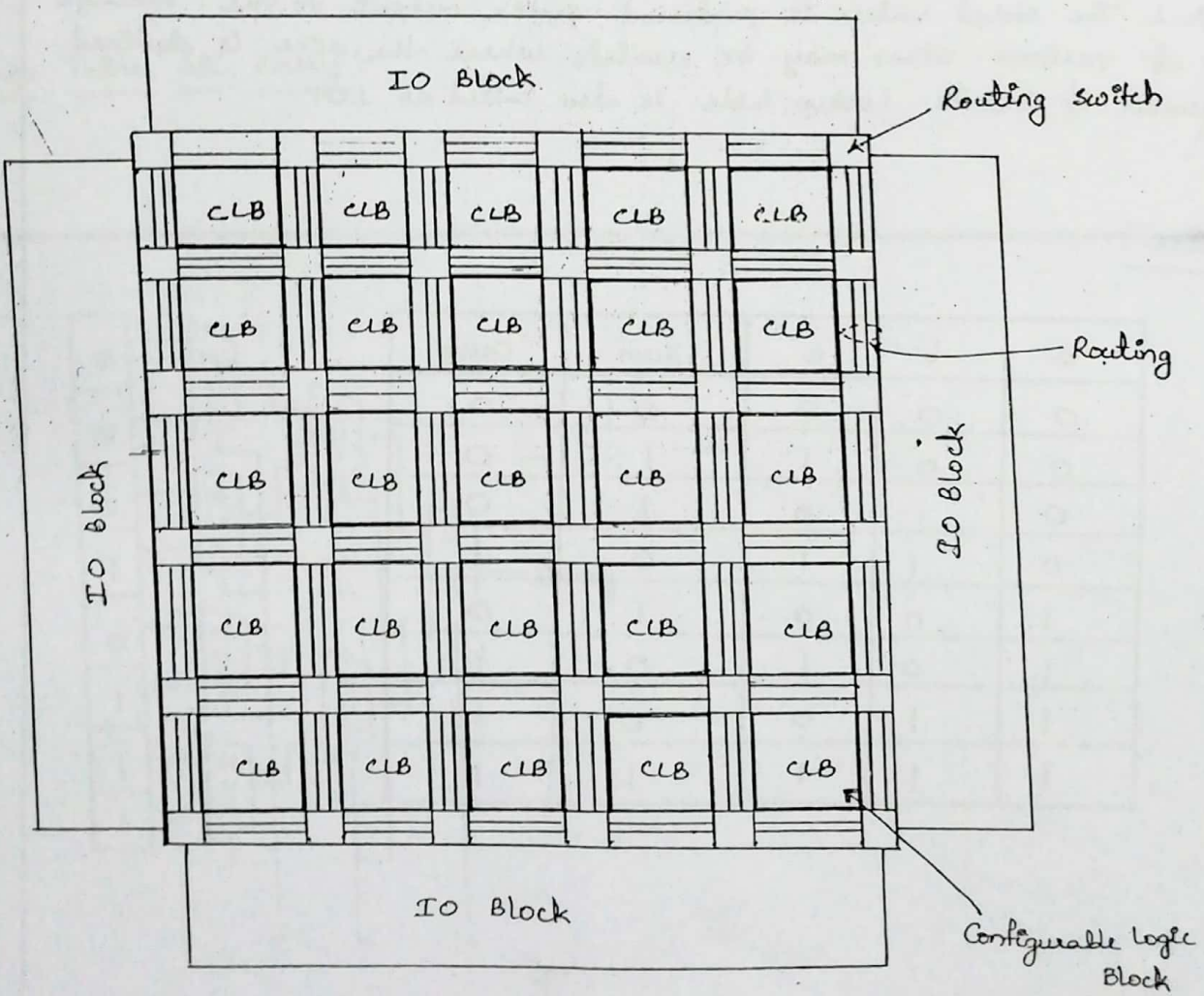
single IC chip depending on the technology used. The design with gate array requires that the customer provide the manufacturer the desired interconnection pattern.

A field programmable gate array (FPGA) is a VLSI circuit that can be programmed in the user's location. A typical FPGA consists of an array of hundreds or thousands of logic blocks, surrounded by programmable input and output blocks and connected together via programmable interconnections.

Programmable Gate Arrays :-

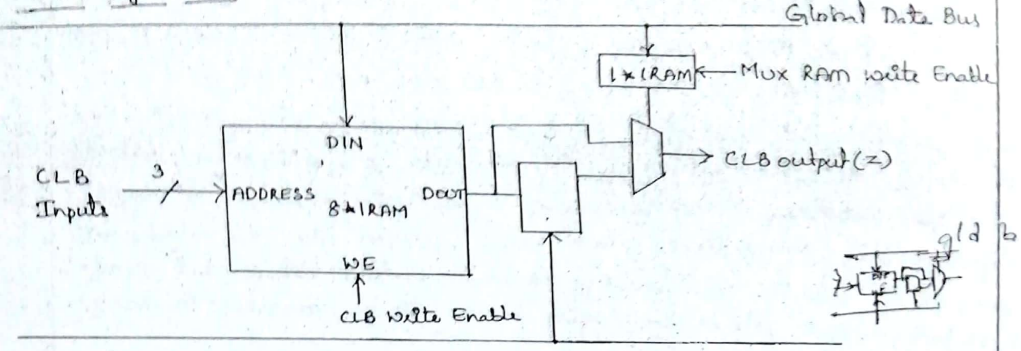
Field Programmable Gate Array (FPGA) is a programmable logic device that supports implementation of relatively large logic circuits. FPGAs can be used to implement a logic circuit with more than 20,000 gates. The general structure of an FPGA contains three main types of resources. They are logic blocks, I/O blocks for connecting to the pins of the package, and interconnection wires and switches. The logic blocks are arranged in a two-dimensional array, and the interconnection wires are organized as horizontal and vertical routing channels between rows and columns of logic blocks. Programmable connections also exist between the I/O blocks and the interconnection wires. The routing channels contain wires and programmable switches that allow the logic blocks to be connected in many ways.

General structure of an FPGA :-



A typical FPGA logic block consists of look-up tables, multiplexers, gates and flip-flops. The look-up table is a truth table stored in a SRAM and evaluates the combinational circuit functions for the logic block.

Simple FPGA logic cell :-

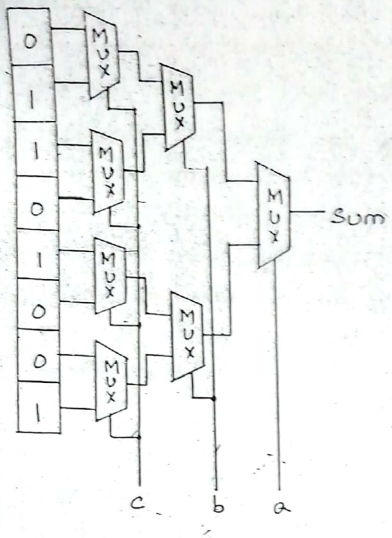


Lookup table contains storage cells that are used to implement a small logic function. Each cell is capable of holding a single logic value, either 0 or 1. The stored value is produced as the output of the storage cell. LUTs of various sizes may be created, where the size is defined by the number of inputs. Lookup table is also called as LUT.

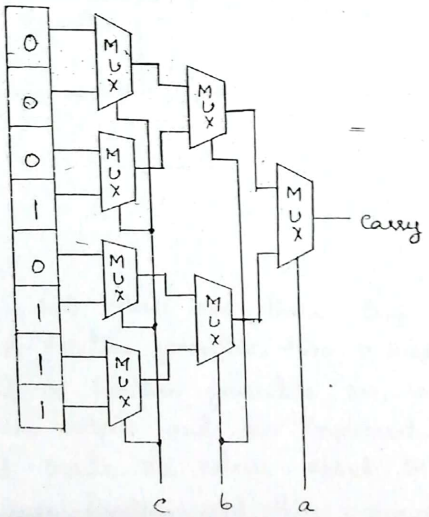
Full adder :-

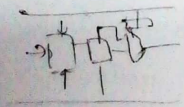
a	b	c	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Table for Sum :-



Look up Table for carry :-





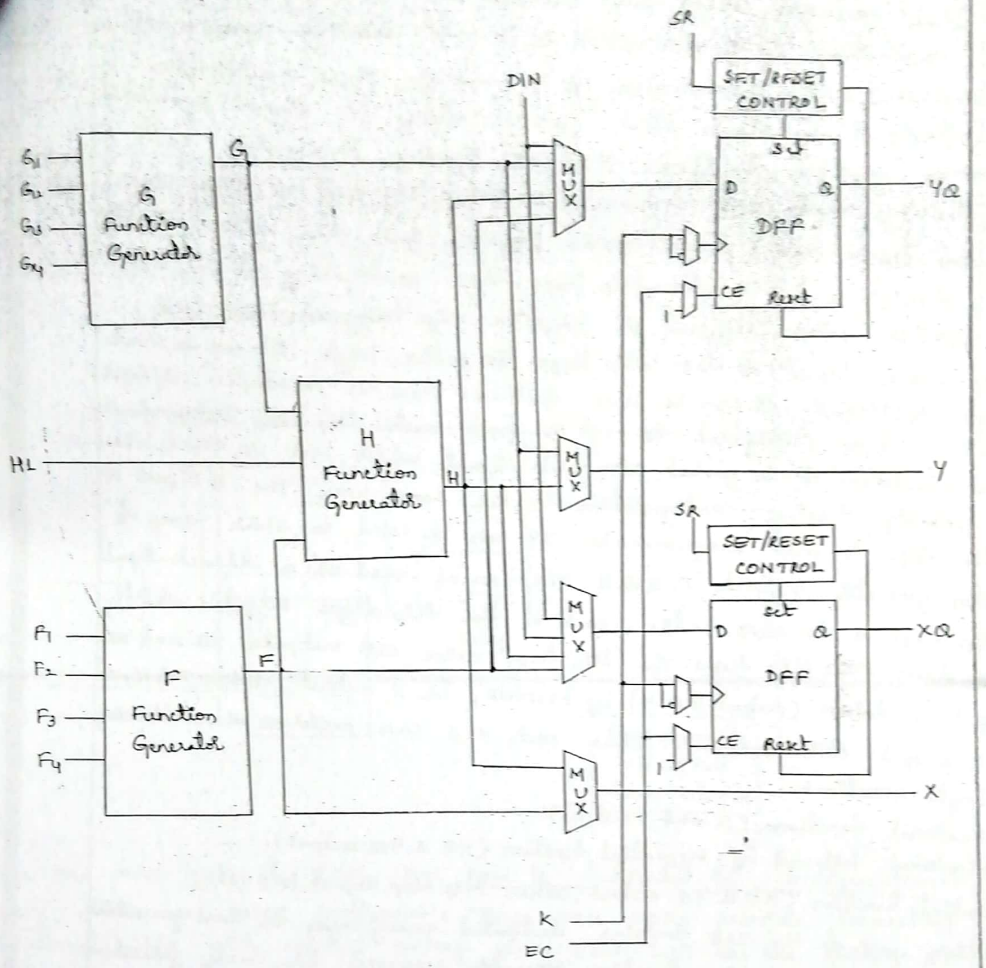
Field programmable Gate arrays use high circuit densities in mo
rocesses to construct ICs. They are completely programmable even after th
re placed in the field. Two basic versions of FPGA are, fuse or anti
ased and SRAM based. First one uses fuse or antifuse to permanen
rogram interconnect and personalize logic. They are one time progra
able. Second one uses small static RAM cells to customize routing
nd logic functions. LUTs are placed in this RAM cells. Compared to
conventional ICs, FPGAs are slower and are less efficient in area
nd power. However, the programming cost is virtually zero. The cost
er part is moderate to high. As initial manufacturing costs for custom
ICs increase and FPGA costs decrease, FPGAs become attractive. Examp
f FPGA based on SRAM technology is XILINX Spartan series FPGA.

XILINX Spartan Series FPGA :-

These FPGAs are implemented with a flexible, regular and reprogr-
ammable architecture of configurable logic blocks (CLBs), interconnected by
programmable routing channels and surrounded by a perimeter of progr-
-ammable input out blocks (IOBs).

FPGAs can be programmed in two modes. They are master serial
mode and slave serial mode. FPGAs can read its configuration data
from an external serial PROM in master serial mode. External device
such as computer can write data into FPGA in slave serial mode.

...able logic block (CLB) :-



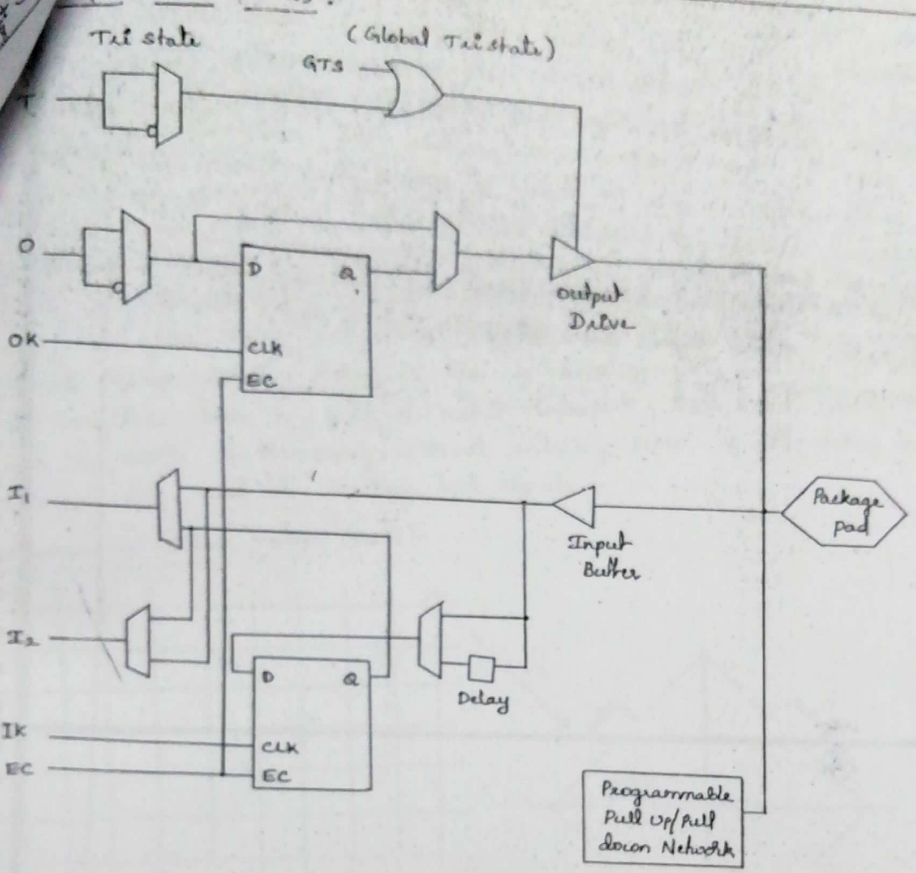
In each CLB there are three look up tables which are used here as F, G and H function generators, two D flip flops and group of multiplexers. Each F and G function generators are 16x1 memory look-up tables having 4-Inputs one output and can Implement any boolean function upto four Independent Inputs. The values stored in memory cells during configuration determine logic function and thus propagation delay is Independent of the logic function realized. H function generator is a three input function generator which can Implement any boolean function of its Input. Two Inputs of H function generator are the outputs from F and G function generator. The other Input (H1) is coming outside of the CLB.

ignals from function generators can exit through two outputs. For H
rough X output and G or H through Y output. Thus a CLB can Implement
y two Input function up to 4 variables each (using F and G), or any
nction of four variables with some function of 5 variables (using F(G)
one and combination of H with G(F)), or it can realize some function
to nine variables (using combination of F, G and H). Thus realization of
ious functions in a single block reduces delay on signal path and
so minimizes number of blocks required, thereby Increasing both
eed and density. Each CLB contains two edge triggered D Flip Flops
ith common clock and clock enable Inputs. Clock can enter directly
- Inverted through a MUX into flip flops, thus configuring D Flip Flops
either positive edge triggered or negative edge triggered. The clock
able / chip enable Input of the flip flops is active high. It can be made
ways high (enable) or it can be user defined. There is another set/Reset
nput which can be configured to set or reset each flip flop Independently
here is a provision of a global set reset signal which set or clear the
lip flops globally during configuration in the same way. The D Input of
here flip flops is also programmable. It can be used to store any of
he function generator outputs F, G or H. DIN can be used as a direct Input
o the flip flops. A 4:1 MUX before each of the flip flops selects which
ignal to drive the flip flops. The flip flops drive CLB outputs XQ and YQ.
Each output is driven (output X or Y) by 2:1 MUX, for X output either F or H,
nd for Y output either G or H. Thus each CLB can perform the following
ypes of logic operations.

(i) Combinational functions (X and Y output)
(ii) Combinational followed by sequential function (XQ & YQ output)
(iii) Sequential function (XQ & YQ output, when Flip Flop Input is DIN).

In addition each of the CLB contains dedicated carry logic for fast generation
f carry signal, therefore saving function generator resources for carry generation

utput Block (IOB) :-



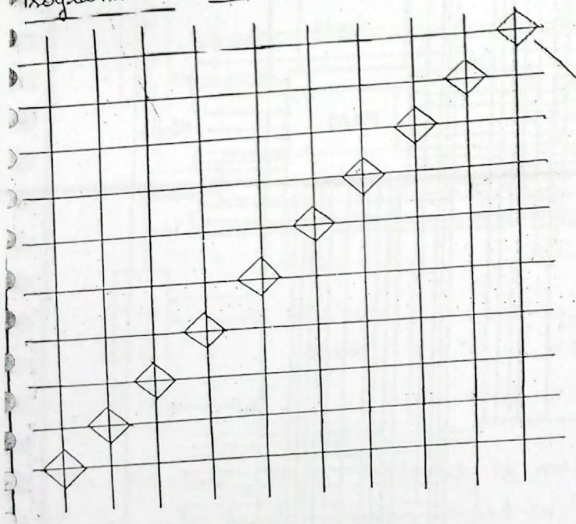
There are two types of IOBs, one type is dedicated for configuration of FPGAs, others are user configurable. These user IOBs provide connection of internal CLBs with external package pins. Each IOB has one package pin which can be configured as input, output & bi-directional signals. The input signal to the IOB can directly go to routing channels and connect CLB inputs & it can go to the register before going to the routing channels. The input register can be configured as an edge sensitive flip flop & level sensitive latch. The clocks can be direct & inverted. This input data path has one tap delay element which can insert partial delay to adjust set up & hold time of flip flops, if necessary. The I1 and I2 signals exist through IOBs each carrying the input either direct & registered. Output signal (O) that is coming from CLB outputs to go as output to the external package pin can be inverted inside IOB, can pass directly to output buffer, & it can be stored in an output register and then pass to the output buffer. Its tri state input, when high places the output buffer

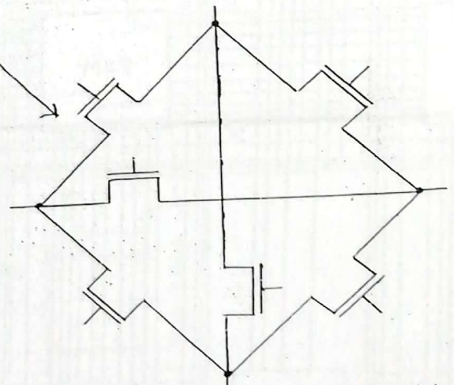
high impedance state. The user controlled T input can go directly or
uted to the tristate control after OR-ing with global tri state (GTS)
nal. GTS signal, which is common to all user IOBs, is made high
ring configuration of FPGA in order to put all the package pins,
cept program pins, in high impedance state. This GTS automatically
es low after configuration is over in order to ensure that FPGA conn-
ects with external circuitary only after configuration is done. Progra-
mable pull-up/pull-down network is used for keeping unused pin to
c & ground level in order to minimize power and increase noise
nsitivity. Seperate clock signals are given to input and output
ip flops. Common enable signal is used.

ling channels around the CLBs have three types of interconnects length, double length and longlines. Each PSM consists of pass istors having programmable gate voltage and during on, establish nection between lines connected at its source and drain. Each SM interconnect point contains six pass transistors in order to route signals in all possible directions.

Single length lines have the greatest interconnect facility since it goes through each PSM block, but due to this phenomenon it also incurs highest amount of delay among the three types of interconnects. Double length lines are twice long than single length lines and they bypass alternate PSMs. So the delay is less but interconnect facility is also less. Longlines form a grid of metal intersect, they run along the length or width of the array, without entering PSMs. So its delay is minimum and used for routing fast signals.

Programmable switch Matrix (PSM) :-





Six Pass Transistors per Switch Matrix Interconnect point

PART A Routing Channels :-

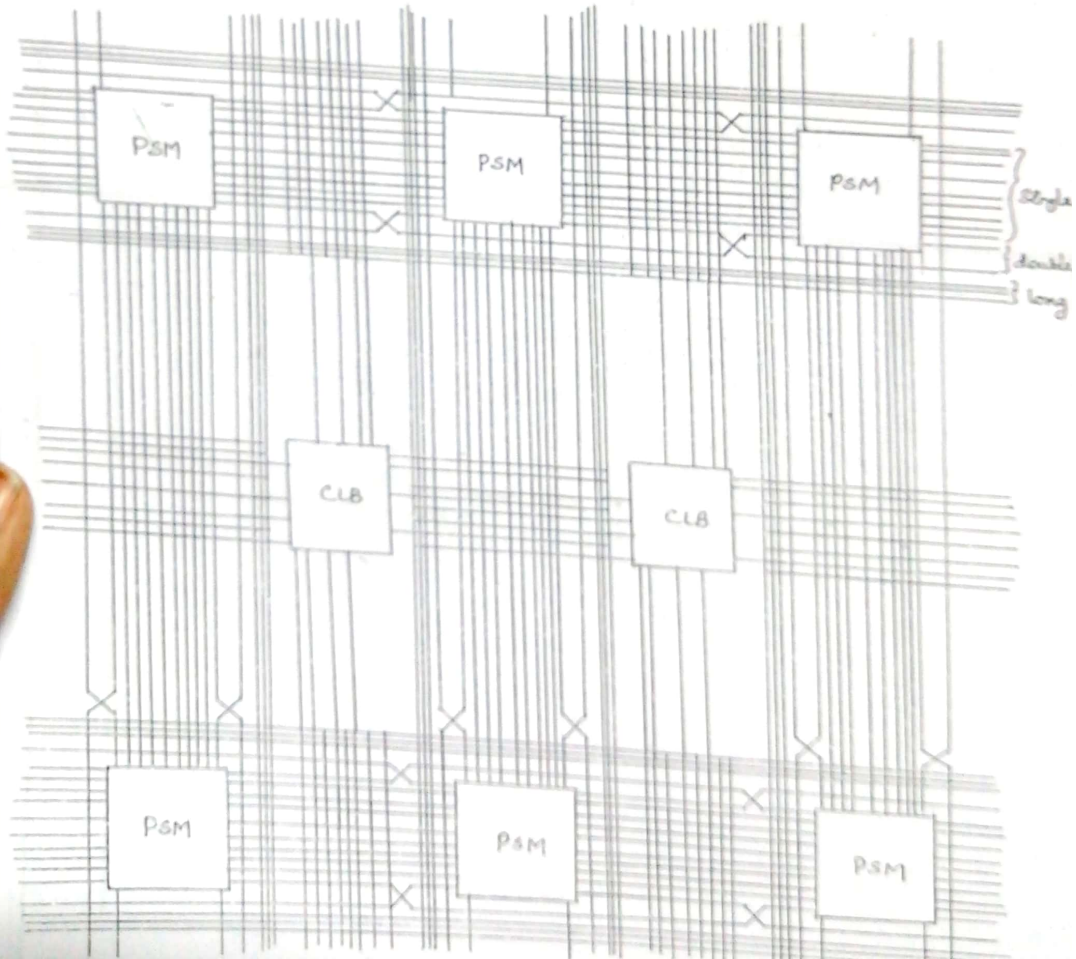
Routing channels, provide paths for interconnecting inputs and outputs of CLBs and IOBs. The internal routing channels are metal segments with programmable pins and programmable switch matrices (PSM).

There are three types of routing channels.

i) CLB routing channels which run along each row and column of CLB array.

ii) IOB routing channels which form a ring, called a Versa ring, around the CLB array. It connects the IOB with the CLB routing channels.

iii) Global routing channels for routing global signals like clocks with minimum delay and skew.



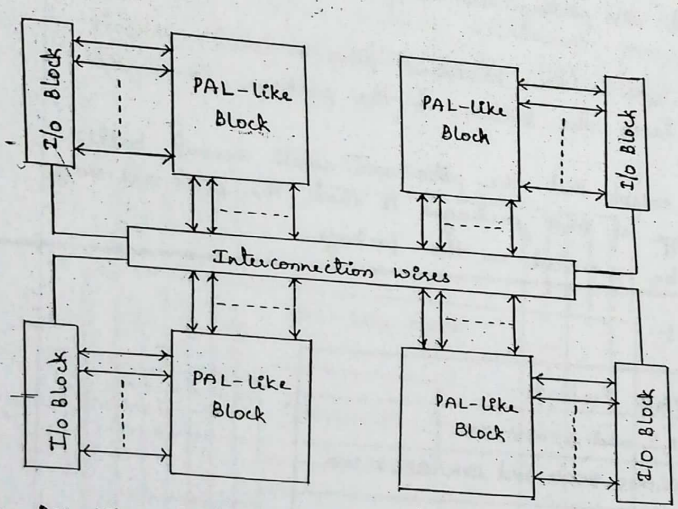
Programmable Logic Devices (CPLDs) :-

... be influen-

... and PALs are useful for implementing a wide variety of small ... circuits. For implementation of circuits that require more inputs ... outputs, either multiple PLAs or PALs can be employed or else a more ... sophisticated type of chip, called a complex programmable logic device, CPLD, can be used.

A CPLD comprises multiple circuit blocks on a single chip, with internal wiring resources to connect the circuit blocks. Each circuit block is similar to a PLA or a PAL. CPLD includes PAL like blocks that are connected to a set of interconnection wires. Each PAL like block is also connected to I/O block. I/O block is connected to input and output pins.

CPLD structure :-



The PAL-like block includes 16 macrocells, each consisting of OR gate, with OR gate inputs between 5 and 20.

A Section of CPLD :-

In the figure below the wiring structure and the connections to a PAL-like are shown. In this example the PAL-like block includes 3 macrocells, each consisting of a four input OR gate. The OR gate output is connected to XOR gate. One input to XOR gate can be programmed to 1 & 0, If 1, then the XOR gate complements the OR gate output, and if 0, then the XOR gate has no effect. The macrocell also includes a flip flop, a multiplexer, and a tristate buffer. The flip flop is used to store the output value produced by the OR gate. Each tristate buffer allows each pin to be used either

FPGA Packaging :-

FPGAs are available in a variety of packages. They are

1) PLCC - Plastic leaded chip carrier.
2) QFP - Quad Flat Pack.
3) PGA - Pin grid array
4) BGA - Ball grid array.

PLCC package has pins that wrap around the edges of the chip on all four sides.

The QFP package pins extend outward from the package. This package has pins on all four sides. The QFP's pins are much thinner than those on a PLCC, which means that the package can support a large number of pins.

A PGA package may have upto a few hundred pins in total, which extend straight outward from the bottom of the package, in a grid pattern.

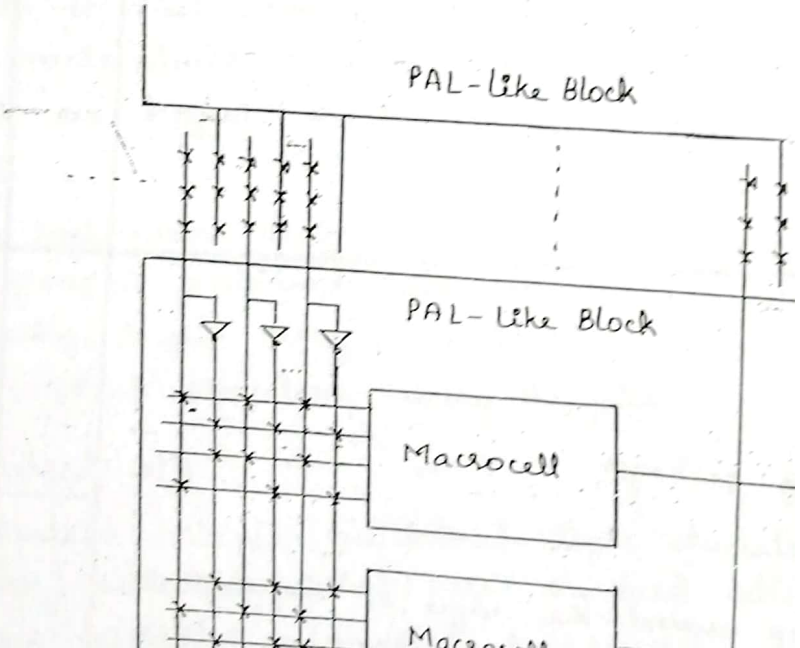
BGA is similar to PGA except that the pins are small round balls, instead of posts. The advantage of BGA packages is that the pins are very small, hence more pins can be provided on the package.

Commercial FPGA products :-

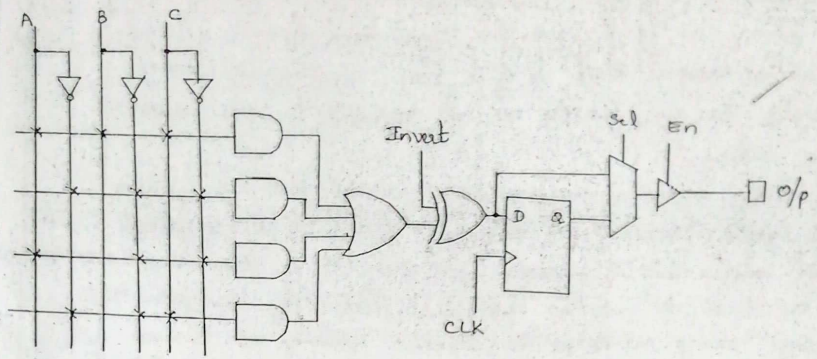
Manufacturer	FPGA Products
Actel	Act 1, 2 and 3, MX, SX
Altera	FLEX6000, 8000 and 10K APEX 20K
Atmel	AT6000, AT40K
Lucent	ORCA 1, 2 and 3
QuickLogic	pASIC 1, 2 and 3
Vantis	VFI
Xilinx	XC3000, XC4000, XC5200, Virtex

output from the CPLD or as an input. responding tri state buffer is enabled, actin If the pin is to be used as an input, then acting as a switch that is turned OFF. In can drive a signal onto the pin, which using the Interconnection wiring.

The Interconnection wiring contains program connect the PAL-like blocks. Each of the hor to some of the vertical wires that it crosses number of switches are chosen to provide su circuits without wasting many switches. Wh as an input, the macrocell associated with th therefore wasted. Some CPLDs include additio macrocells and the interconnection wiring th in such situation.



Macrocell :-



$F = ABC + \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + \bar{A}\bar{B}C$

En	Invert	Sel	output
0	x	x	z
1	0	0	F
1	0	1	delayed F
1	1	0	F'
1	1	1	delayed F'

CPLD Packaging :-

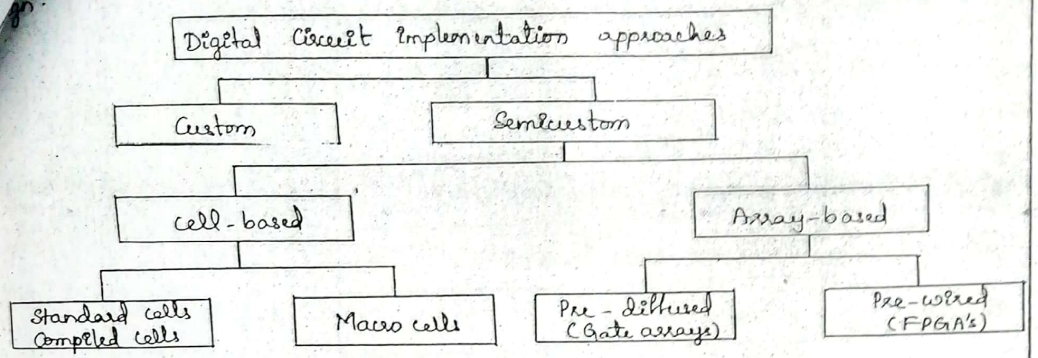
CPLDs are available in the following packages.

1) Plastic leaded chip carrier - PLCC
2) Quad flat pack - QFP

The PLCC package has pins that wrap around the edges of the chip on all four sides. The socket that houses the PLCC is attached by solder to the circuit board, and the PLCC is held in the socket by friction. The QFP package has pins on all four sides, and they extend outward from the package, with a downward-wiring shape.

Approaches :-

llowing figure shows various implementation approaches of VLSI gn.



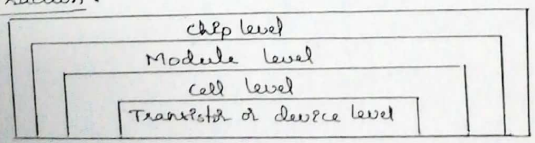
In the custom design approach, each individual transistor is designed and laid out manually. The main advantage of this method is that the circuit is highly optimized for speed, area & power. This approach is also known as full custom design approach. In a full custom design approach the semiconductor chips are Application Specific Integrated Circuits (ASICs) which are designed specifically for a given application or application domain.

In semi custom design approach, the majority of the chip is designed using a group of predefined cells called as standard cells and rest are designed manually. A gate array is an IC chip on which gates are placed in matrix form (without connections among the gates.

Standard cells :-

Standard cells are pre defined logic elements used in the circuit. The design methodology that uses standard cells is known as cell based design methodology. A standard cell library is one of the foundations upon which the VLSI design approach is built. A standard cell is designed either to store information & perform a specific logic function. The type of standard cell created to store data is referred to as a sequential cell. Flip flops and latches are examples of sequential cells. The type of standard cell used to perform logic operations on the signals presented on its inputs is called combinational cell.

Level of abstraction :-



Commercial CPLD Products :-

Manufacturers	CPLD Products
Altera	Max 5000, 7000 and 9000
Atmel	ATF, ATV
Cypress	FLASH 370, Ultra 37000
Lattice	Esp LS1 1000 to 8000
Philips	XPLA
Vantis	MACH 1 to 5
Xilinx	XC 9500

Comparision of FPGA and CPLD :-

FPGA	CPLD
1) FPGA architecture contains gate array like structures.	1) CPLD architecture contains PAL like structures.
2) Its density range is from medium to high.	2) Its density range is from low to medium.
3) FPGA speed depends on the application.	3) CPLD speed is very high.
4) For Interconnection purpose routing channels are used.	4) For Interconnection purpose crossbar connections are used.
5) In FPGA power consumption is medium.	5) In CPLD power consumption is high.
6) Once supply is removed, data is lossed in FPGA, i.e. they are volatile.	6) CPLD contains on chip non-volatile memory.
7) FPGAs are Internally based on look-up tables (LUTs).	7) CPLDs form the logic functions with sea-of-gates.

Factors Influencing low power VLSI Design :-

The overall power consumption of an integrated circuit can be influenced at all levels of its design fabrication technology, circuit optimization, logic design, control and clocking strategies, architectural partitioning and layout, and the underlying system's algorithm. To reduce the power consumption different methods are available.

V_{DD} :-

The first strategy is to reduce V_{DD}. Variation in V_{DD} leads to quadratic change in the power delay product.

Threshold voltage :-

By reducing V_t to lower levels it would be possible to reduce V_{DD} even further.

Impensating for lower speed :-

As the industry moves from the standard 5V process to ones with lower supply voltages, design engineers need to compensate for the loss of performance. There are two architectural approaches, first apply the standard speed optimisation techniques, second use parallelism. Pipelining is the standard technique for increasing the overall speed of a circuit. The idea of using parallelism is simply to have more operations being conducted at the slower speed to achieve the same overall performance.

Voltage swing :-

A final way of reducing power loss connected with supply voltage is to reduce voltage swing.

Reduce C :-

The second strategy is to reduce capacitance. This comes naturally with smaller feature sizes.

Partition blocks :-

As a general rule it is best to partition large blocks into smaller ones. Power calculation for each block depends on the capacitance of that block. When large blocks are partitioned into smaller blocks, only one block is operated at a time due to this product of activity and capacitance is reduced.

Clocks and Control :-

In architectures with distributed processing there should be global control and clock signals, but the global distribution network has a very large capacitance and is switched frequently. To overcome this several