UNIT-V

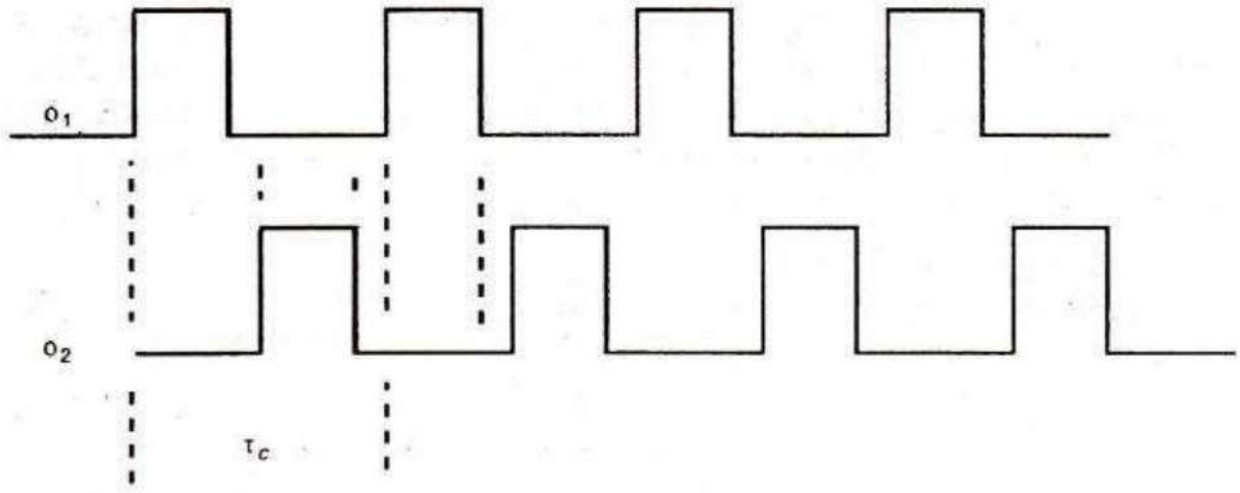
Subsystem Design and Layout-2: Clocked sequential circuits, dynamic shift registers, bus lines, General considerations, 4-bit arithmetic processes, 4-bit shifter, Regularity Definition& Computation Practical aspects and testability: Some thoughts of performance, optimization and CAD tools for design and simulation.

****

Clocked sequential circuits

Two-Phase Clocking:

The clocked circuits to be considered here will be based on a two-phase non- overlapping clock signal as defined by Figure.



Notes: 1. τ_c = clock period
2. $o_1(t)$. $o_2(t) = 0$; *all* t

- A two-phase clock offers a great deal of freedom in sequential circuit design if the clock period and the duration of the signals PHI-1 and PHI-2 are correctly chosen.
- If this is the case, data is allowed to become stable before any further transfer takes place and there is no chance of race conditions occurring.
- Clocked circuitry is considerably easier to design than the corresponding asynchronous sequential circuitry. It does, however, usually pay the penalty of being slower. However, at this stage of learning VLSI design we will concentrate on two-phase clocked sequential circuits alone and thus simplify design procedures.
- When studying Figure 6.31, it is necessary to recognize the fact that PHI-1 and PHI-2 do not need to be symmetrical as shown.
- For a given clock period, each clock phase period and its. associated under lap period can be varied if the need arises in optimizing a design.

- A very simple arrangement using combinational logic and generating a two- phase clock at the frequency of a single-phase input clock is set out in Figure 6.33(a).

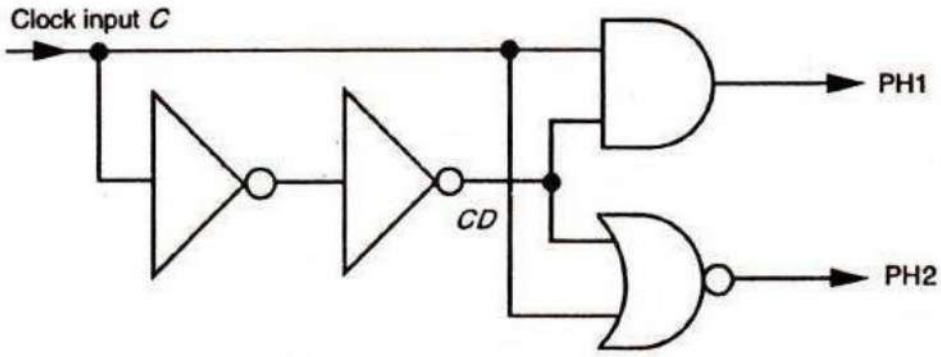


FIGURE 6.33(a) Simple two-phase clock generator circuit—basic form.

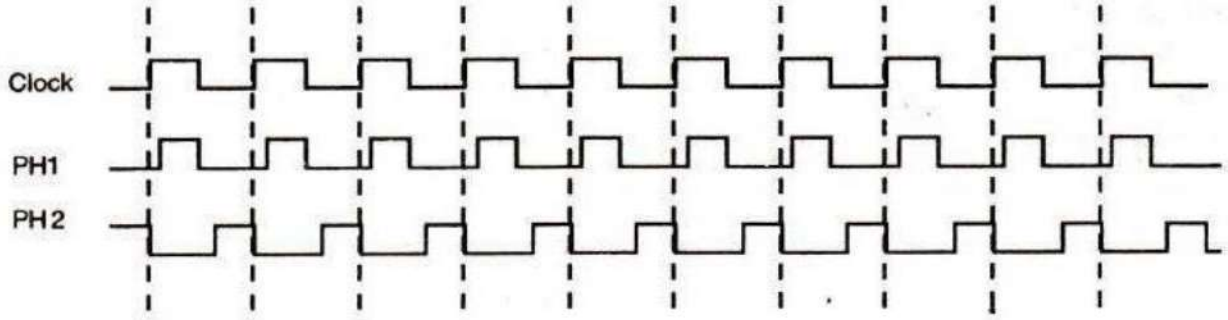


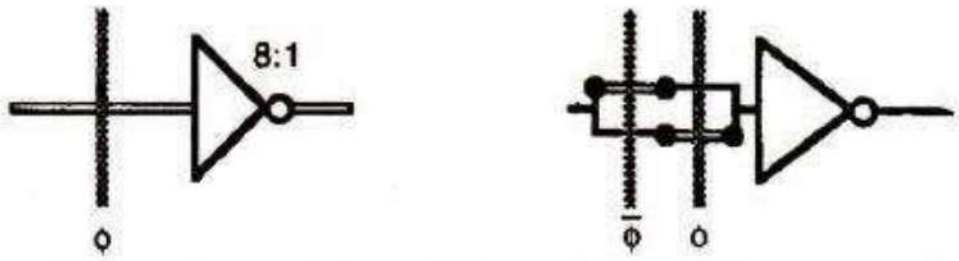
FIGURE 6.33(b) Waveforms for two-phase clock generator.

- The input clock signal C is used to provide a delayed version of it (CD) by passing it through an even number of inverters. The delay thus produced determines the underlap period for the two phase clock.
- Waveforms are as shown in Figure 6.33(b). The phase 1 signal PHI-1 is generated by ANDing C with CD whilst the phase 2 signal PHI-2 is produced by NORing C with CD (that is, ANDing C' with CD'). Clearly, the minimum underlap period will be that generated by the delay through two inverters and this is also the increment by which the delay may be increased by adding further inverter pairs.

Dynamic Shift Registers

Dynamic Register Element

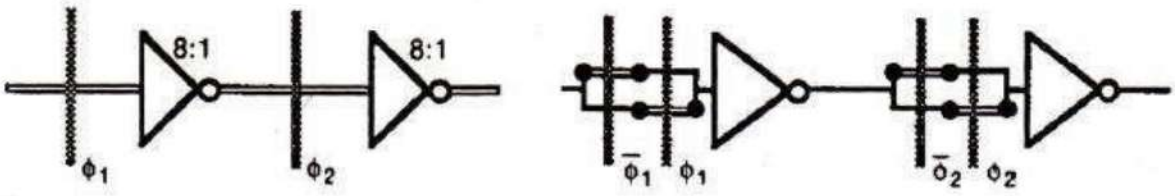
The basic dynamic register element is shown in Figure 6.36 in mixed stick/circuit notation and may be seen to consist of three transistors for nMOS and four for CMOS per stored bit in complemented form. The element's operation is simple to appreciate. (Vin) is clocked in by PHI-1 (or PHI-2) of the clock and charges the gate capacitance Cg of the inverter to Vin



(a) nMOS pass transistor switched (b) CMOS transmission gate switched

FIGURE 6.36 Basic inverting dynamic storage cells.

- If uncomplemented storage is essential, the basic element is modified as indicated in Figure 6.37 and will be seen to consist of six transistors for nMOS and eight for CMOS. Data clocked in on PHI-1 is stored on Cg1 and the corresponding output appears at the output of inverter 1. On PHI-2 this value is clocked into and stored by Cg2 and the output of inverter 2 then presents the 'true' form of the stored bit. Note that data read in on PHI-1 is not available at the output until sometime following the next positive edge of the clock signal PHI-2.



(a) nMOS pass transistor switched (b) CMOS transmission gate switched

FIGURE 6.37 Non-inverting dynamic storage cells.

4-bit Dynamic Shift Register

- Cascading the basic elements of Figure 6.37 gives a serial shift register arrangement which may be extended to *n* bits. A four-bit serial right shift nMOS register is illustrated in Figure 6.38(a).
- Data bits are shifted in when ¢1.LD is present, one bit being entered on each ¢1signal (provided that *LD* is logic 1). Each bit is stored in Cg1 as it is entered, and then transferred complemented into *Cg2* during the next ¢2. Thus, after a ¢1 followed by ¢2 signal, the stored bit is present at the output of inverter 2.
- On the next ¢1 the next input bit is storedin Cg1 and simultaneously the first bit stored is passed on to inverter pair 3 and 4 by being stored in *Cg3,* and so on.
- It will be seen that bits are thus clocked to the right along the shift register on each ¢1 followed by ¢2 sequence.
- Once four bits are stored, the data is available in parallel form at the outputs of inverters 2, 4, 6 and 8, and is also available in serial form from the output of inverter 8 when ¢ii.RD is high as further clock sequences are received (where *RD* is the serial read control signal).

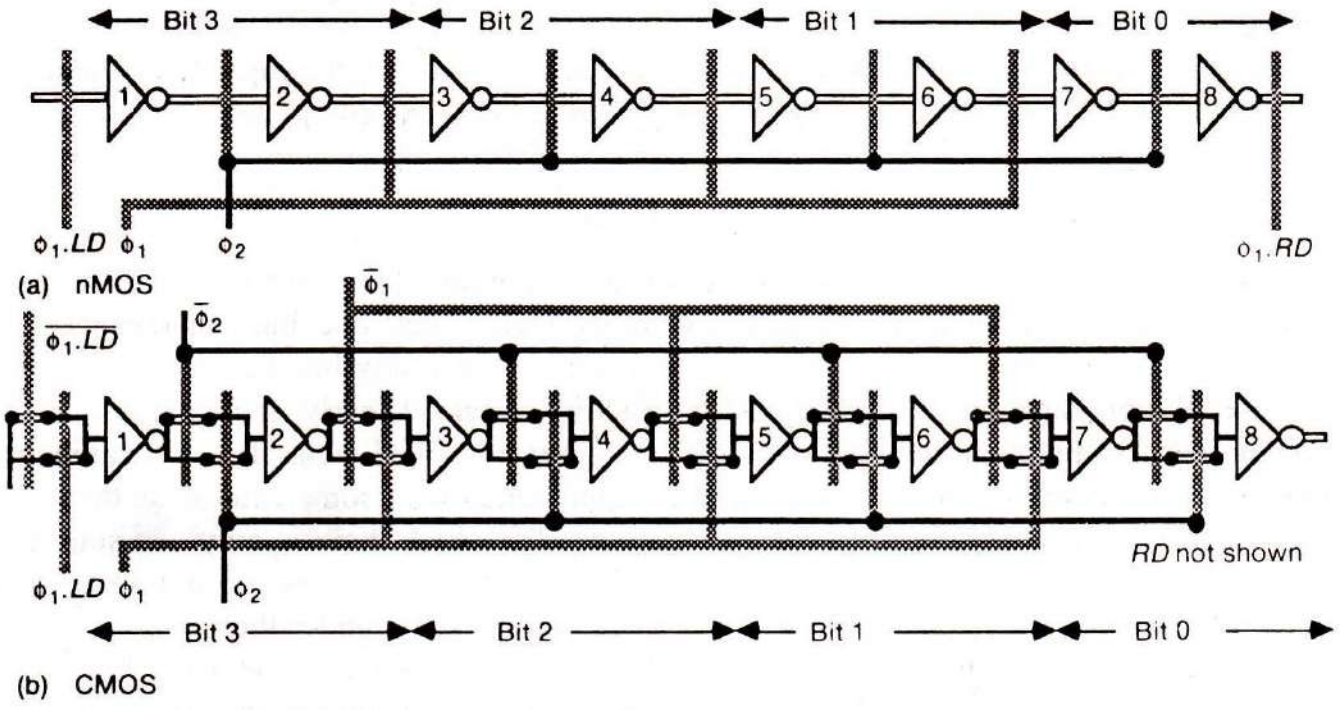


FIGURE 6.38 Four-bit dynamic shift registers (nMOS and CMOS).

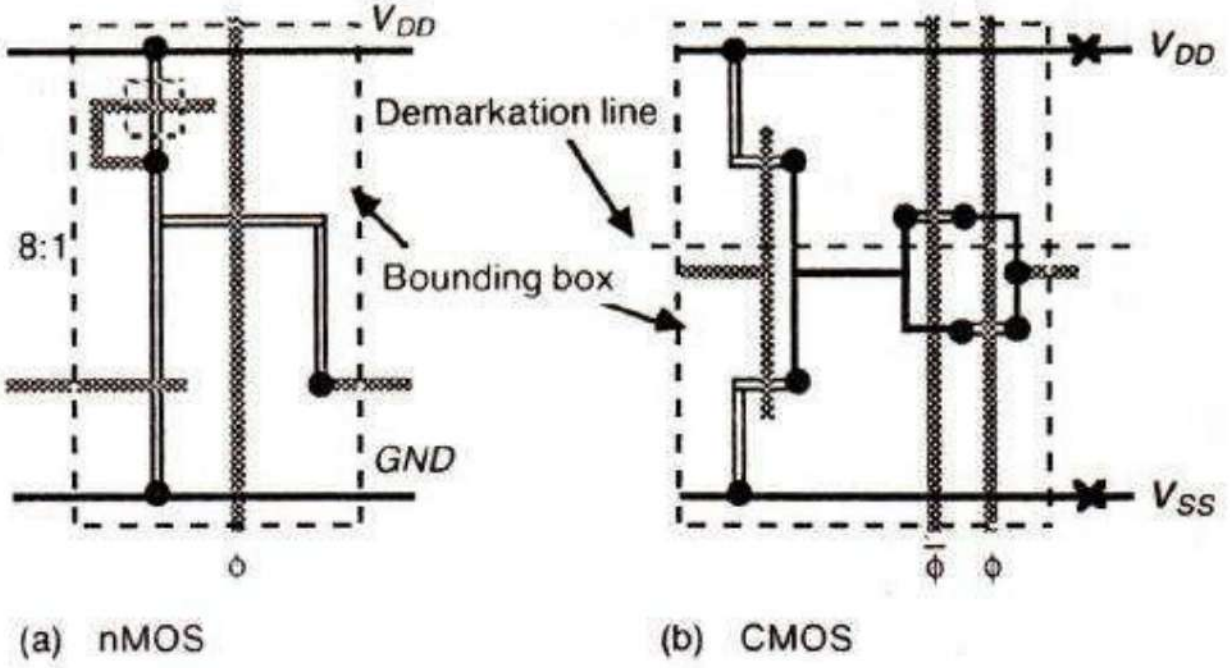


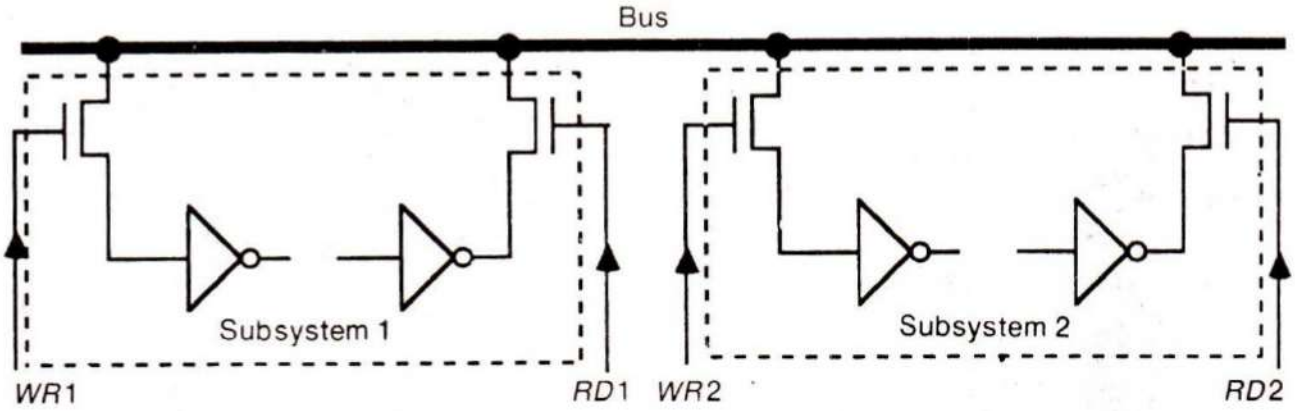
FIGURE 6.39 Stick diagrams for shift register cells.

BUS Lines

When designing at leaf-cell level, it is easy to lose sight of overall system requirements and restrictions. In particular, the use of buses to interconnect subsystems and circuits must always be most carefully considered; such matters and the current-carrying capacity of aluminum wiring used for V_{DD} and *GND* or *Vss* rails are often overlooked completely.

There are three classes of bus-passive, active, and precharged. A *passive* bus rail is a floating rail to which signals may be connected from drivers through series switches, for example, pass transistors, to propagate along the bus and from which signals may be taken, also through pass transistors.

A form of *active* bus is to treat the bus rail as a wired *Nor* connection which has a common pull-up *Rp.u.* and n-type pull-down transistors or series n-type transistor logic pull-downs where there are circuits which must be selected to drive the bus. Signals are taken off the bus in a similar manner and the general arrangement is given as Figure 6.42. This arrangement is not suited to complementary CMOS logic-based designs since it is based on pull-down logic only.



Note: For CMOS the pass transistors could become transmission gates.

FIGURE 6.41 Passive bus—nMOS or CMOS.

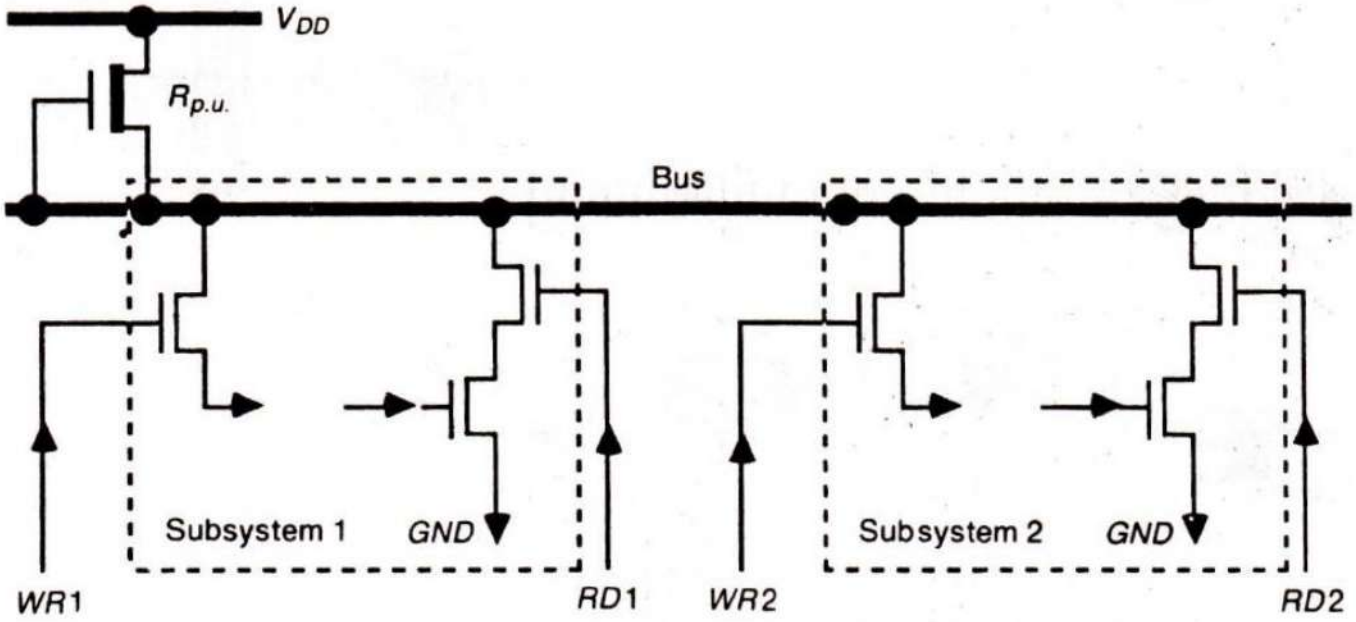


FIGURE 6.42 Active bus (not CMOS).

The passive bus suffers from ratio problems in that, for any reasonable area restrictions on the bus driver circuits, the bus will be slow to respond, particularly for the ~V (logic 0 to 1) transitions, because of

the relatively high value pull-up resistance of the drivers and the associated series pass transistor or transmission gate.

The active bus is better in that more time is available for the bus to charge to *V DD*• since *Rp.u.* is always connected to the bus and there are no series pass transistors between *Rp.u.* and the bus. However, there are still ratio problems which limit the speed of the bus if reasonable area is to be occupied.

The *precharged* bus approach limits the effects of bus capacitance in that a single pull-up transistor which is turned on only during Ø2 (say) provides for the bus to charge during the Ø2 on period; the size of this transistor can be made relatively large (i .e. a low *L: W* ratio) and, therefore, have a low resistance. There are no ratio problems between it and the bus drivers since they are never turned on at the same time. The bus drivers merely pull down (or not) the precharged bus by discharging *Cg.* The arrangement is given at Figure 6.43 and, in effect, a ratioless precharged wired *Nor* circuit is formed by the bus system. However, care must be taken in nMOS systems when using logic 1 levels from the bus since the bus never reaches *VDD*• due to threshold voltage effects in the precharging transistor.

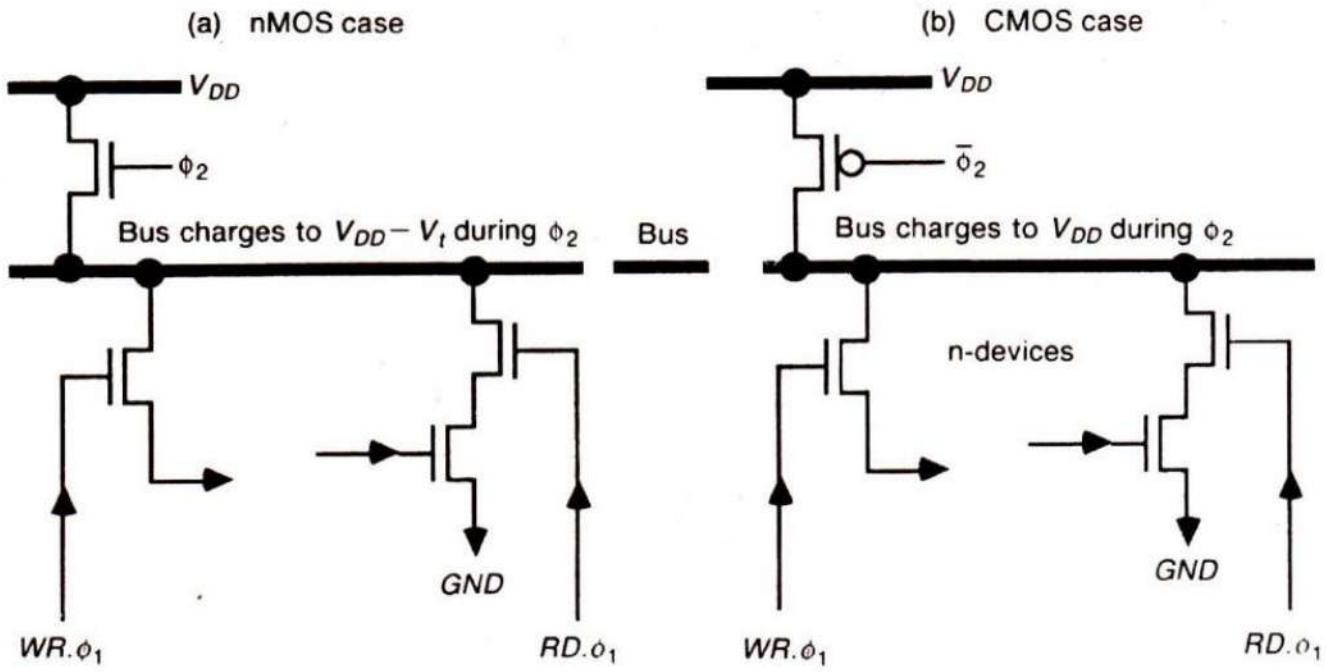


FIGURE 6.43 Precharged bus—nMOS and CMOS.

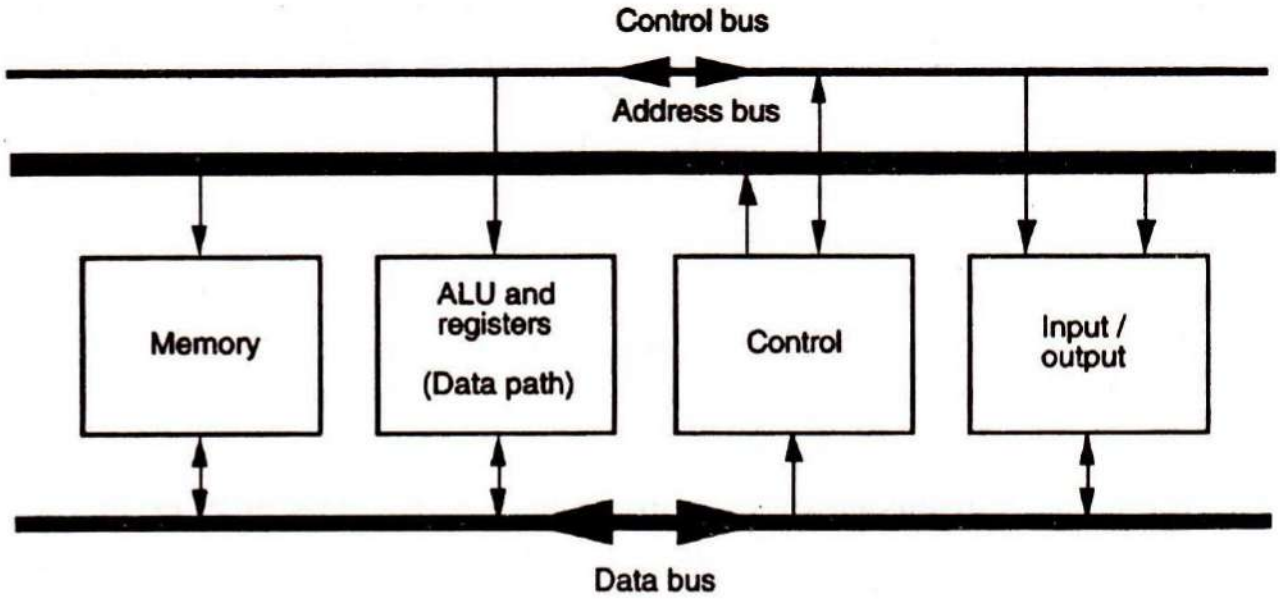
General considerations

1. *Lower unit cost* compared with other approaches to the same requirement.
2. *Higher reliability.* High levels of system integration usually greatly reduce interconnections.
3. *Lower power dissipation, lower weight, and lower volume* compared with most other approaches to a given system
4. *Better peiformance-particularly* in terms of speed power product.
5. *Enhanced repeatability.* There are fewer processes to control if the whole system or a very large part of it is realized on a single chip.

6. *The possibility of reduced design/development periods* (particularly for more complex systems) if suitable design procedures and design aids are available.

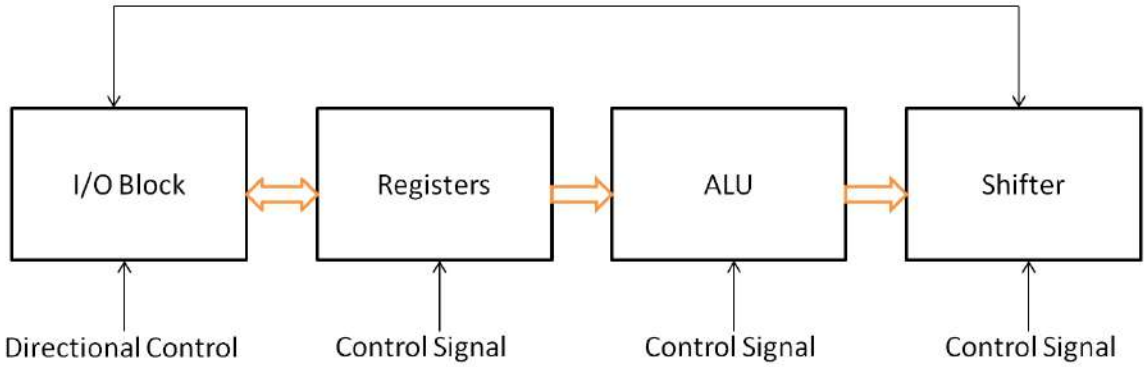
4 bit arithmetic process

4 bit microprocessor is design with different stages.



Basic digital processor architecture

Design of data path circuits:

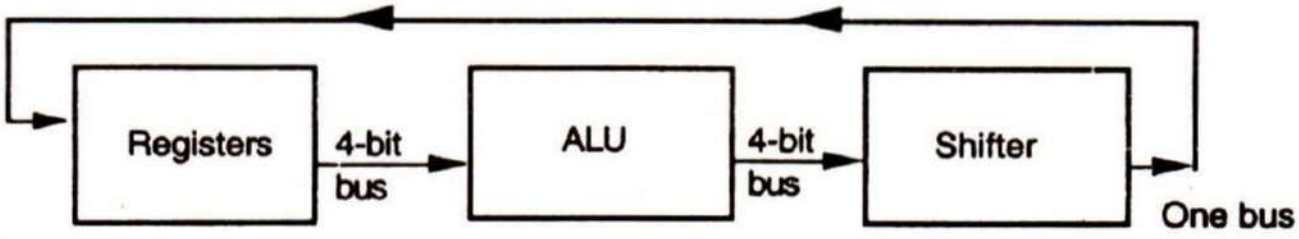


Data path block consists of arithmetic operations, logical operations, shift operations and temporary storage of operands.

Types:

1. One bus architecture
2. Two bus architecture
3. Three bus architecture

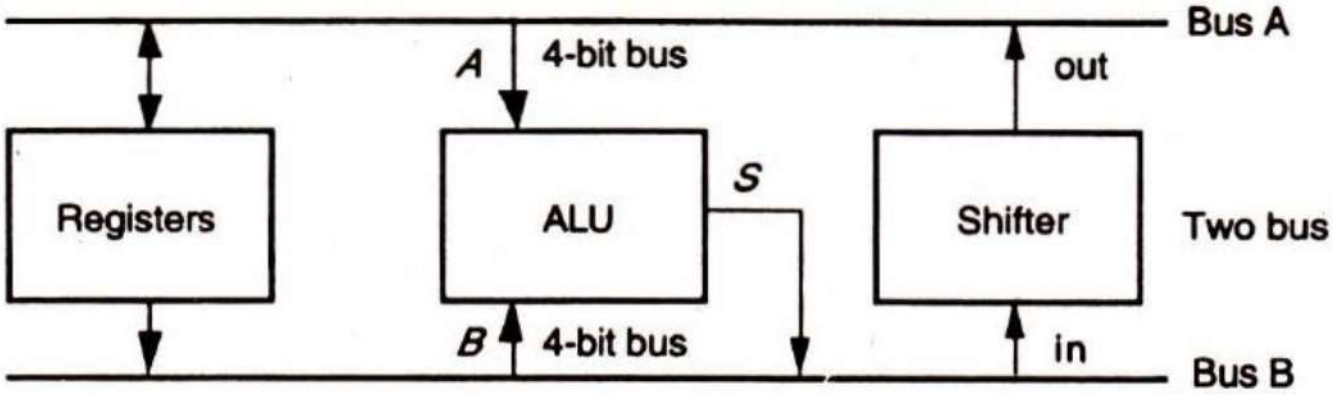
One bus architecture



Sequence:

1. 1st operand from registers to ALU. Operand is stored there.
2. 2nd operand from register to ALU and added.
3. Result is passed through shifter and stored in the register

Two Bus Architecture

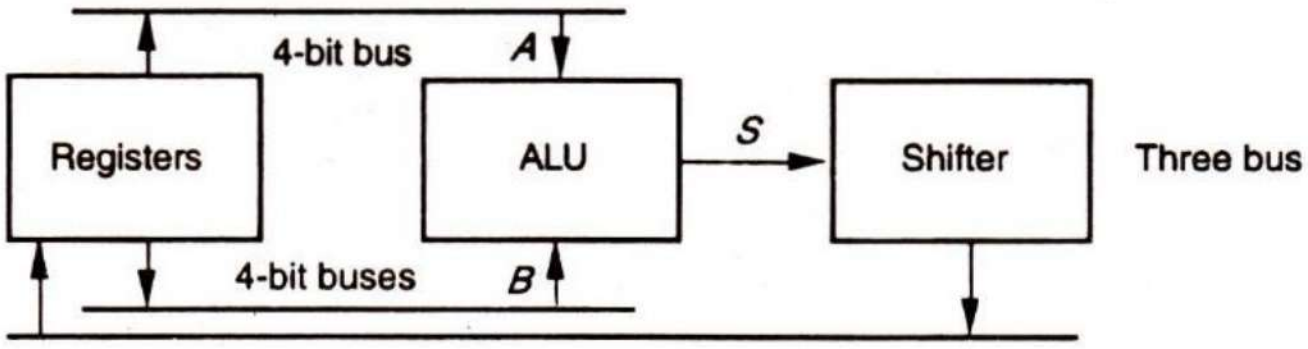


Sequence:

1. Two operands (A & B) are sent from register(s) to ALU & are operated upon, results in ALU.
2. Result is passed through the shifter & stored in registers.

Three bus architecture:

Two operands are moved from the register to ALU corresponding operation is performed. Result is passé through shifter block and move to data bus. Same clock period but it uses three bus architecture.



Shifters:

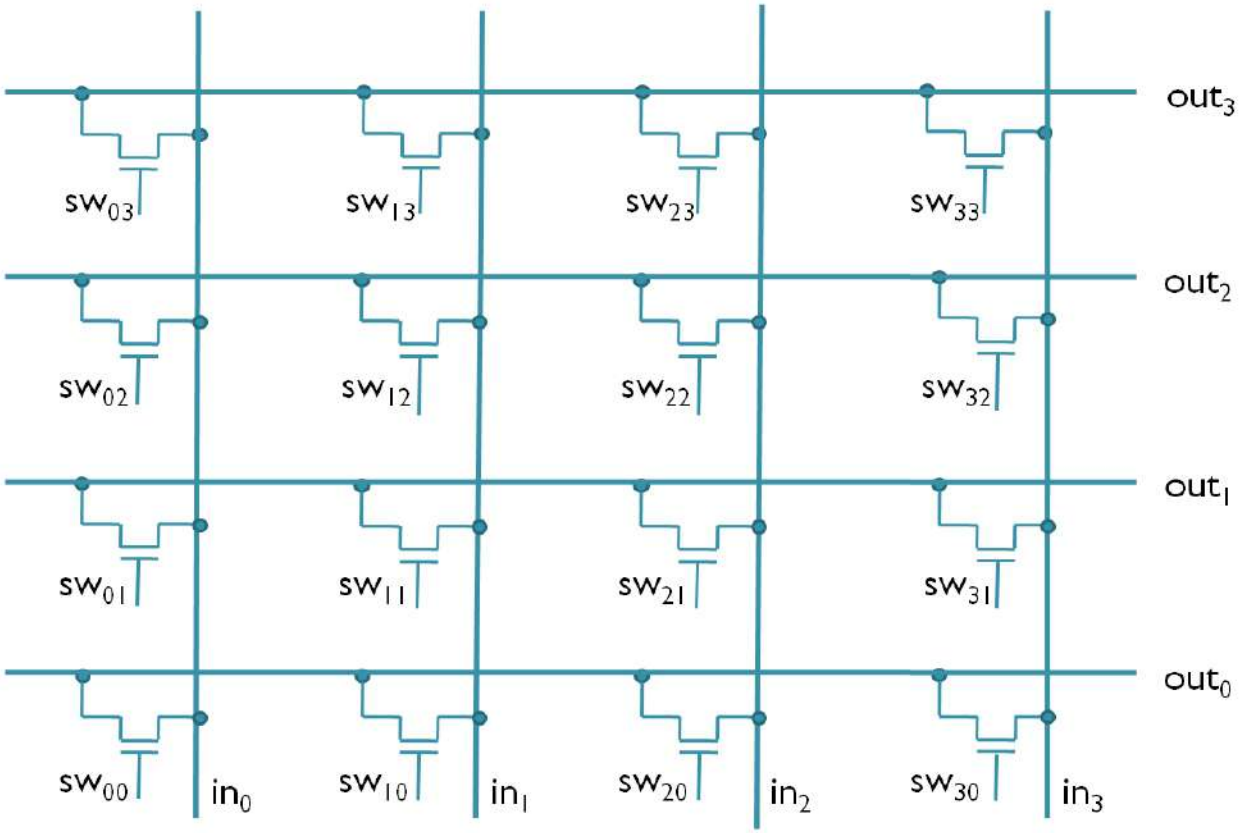
- Shifters are important elements in microprocessor designs for arithmetic shifting, logical shifting and rotation functions.
- The shift register is one of the simplest shifter that can shift one position for clock cycle.
- In Binary Operation, shifting is a bitwise operation that shifts the operand 1 to n-1 places to the right or left.

Design of a 4-bit shifter

Any general purpose n-bit shifter should be able to shift incoming data by up to n – 1 place in a right-shift or left-shift direction. Further specifying that all shifts should be on an end-around basis, so that any bit shifted out at one end of a data word will be shifted in at the other end of the word, then the problem of right shift or left shift is greatly eased. It can be analyzed that for a 4-bit word, that a 1-bit shift right is equivalent to a 3-bit shift left and a 2-bit shift right is equivalent to a 2-bit left etc. Hence, the design of either shift right or left can be done. Here the design is of shift right by 0, 1, 2, or 3 places.

The shifter must have:

- input from a four line parallel data bus
- four output lines for the shifted data
- means of transferring input data to output lines with any shift from 0 to 3 bits



- 4 x 4 crossbar switch

Consider a direct MOS switch implementation of a 4 X 4 crossbar switches shown in figure. The

arrangement is general and may be expanded to accommodate n-bit inputs/outputs. In this arrangement any input can be connected to any or all the outputs.

Furthermore, 16 control signals (sw00 – sw33), one for each transistor switch, must be provided to drive the crossbar switch, and such complexity is highly undesirable.

In case of ordinary 4 x 4 shifter we need 16 control inputs from shift 00 to shift 33, 4 data inputs $a_3a_2a_1a_0$ supply through input rails in_0 to in_3 and output signals $f_3f_2f_1f_0$ dragged at the out_3 to out_0 as shown in figure.

The output of 4 x 4 ordinary shifter without any shift operation

$$f_3f_2f_1f_0= a_3a_2a_1a_0$$

If one bit left shift rotation yields

$$f_3f_2f_1f_0= a_2a_1a_0a_3$$

If one bit right shift rotation yields

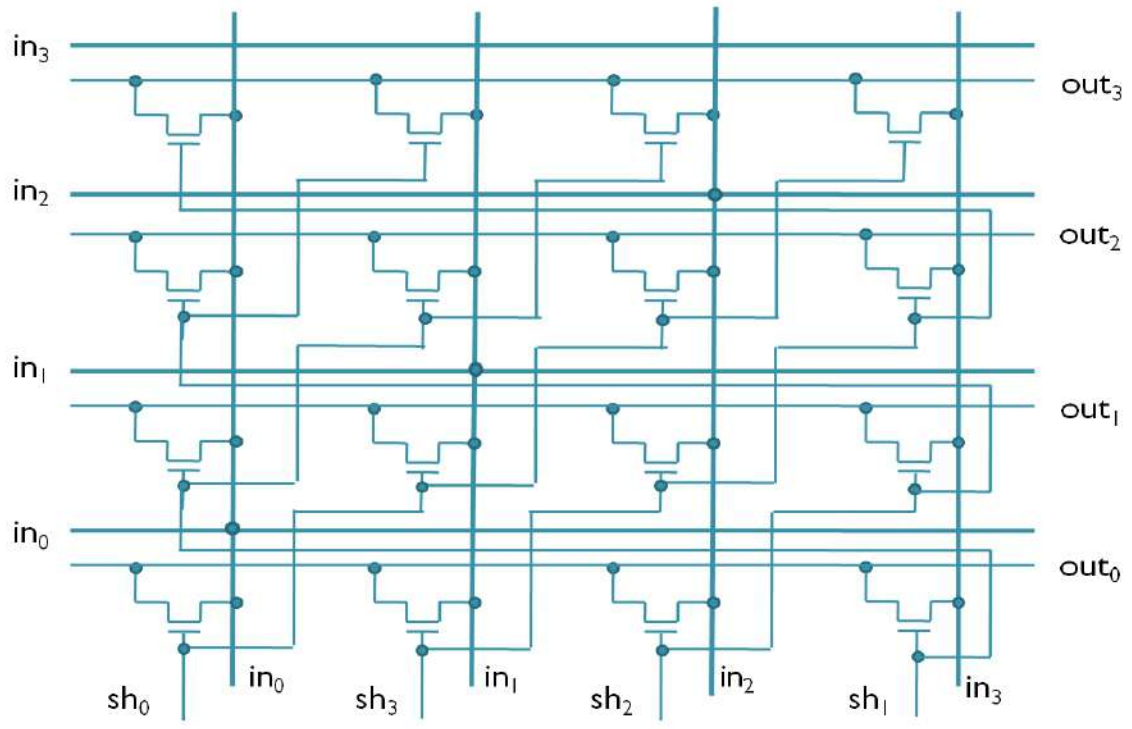
$$f_3f_2f_1f_0= a_0a_3a_2a_1$$

Ex: for one bit left rotation the control signals SW03, SW01, SW21,SW23 are said to one and remaining all control inputs are said to zero.

Note: Due to disadvantages of number of control inputs to run a ordinary shifter we go for barrel shifter with number of control inputs equal to number of input bits.

Barrel Shifter

A barrel shifter is a very efficient layout, it can perform n-bit shifts in a single combinational function. Consider a 4 bit word $a_3a_2a_1a_0$ is input to the barrel shifter and $f_3f_2f_1f_0$ taken as output signals. The N-bit rotation is specified by using the control word sw_n as shown ion figure



Usually a transmission gate is core of cell which can built from a single N type transistor. The input data runs diagonally upwards through the system, the output data runs horizontally the control signals runs in stair case fashion as shown in figure.

Regarding to the barrel shifter the important consideration is selection control signals combination.

If the actual input pair is

$$f_3f_2f_1f_0 = a_3a_2a_1a_0$$

for one bit left rotation we have to change the control inputs accordingly. Any one of the control signal is said to be one while remaining control signals are said to zero. After one bit left rotation outputs

$$f_3f_2f_1f_0 = a_2a_1a_0a_3$$

For this operation the combination of control signals should be

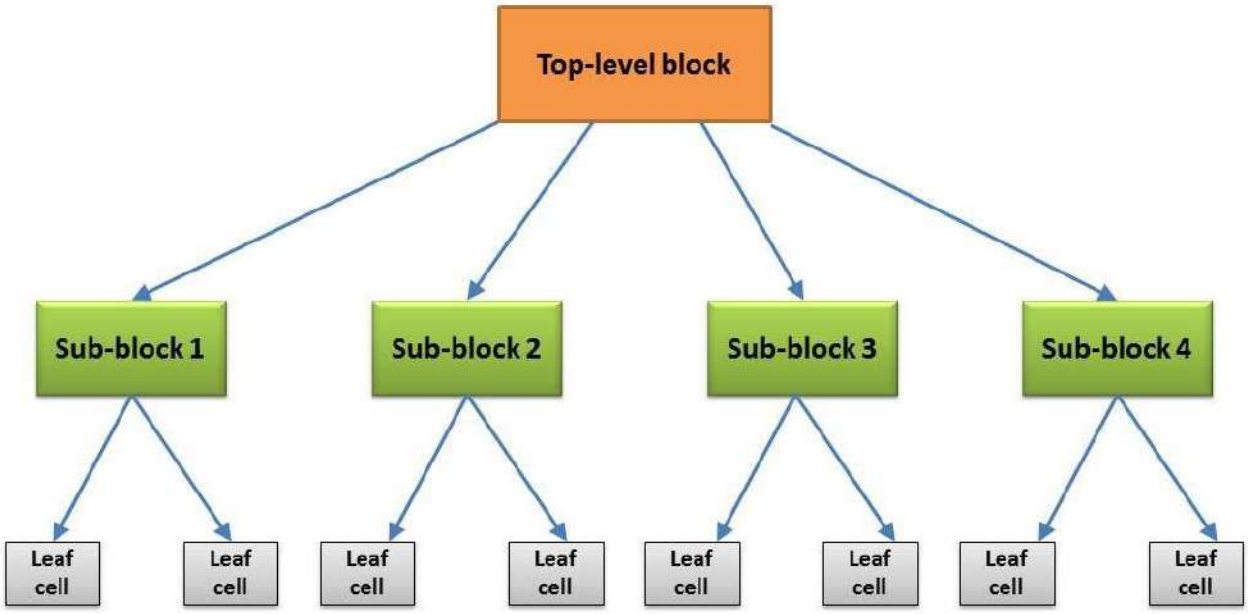
$$Sh_0\ Sh_1\ Sh_2\ Sh_3 = 0\ 0\ 0\ 1$$

The inter bus switches have their gate inputs connected in a staircase fashion in groups of four and there are now four shift control inputs which must be mutually exclusive in the active state. CMOS transmission gates may be used in place of the simple pass transistor switches if appropriate. Barrel shifter connects the input lines representing a word to a group of output lines with the required shift determined by its control inputs (sh0, sh1, sh2, sh3). Control inputs also determine the direction of the shift. If input word has n – bits and shifts from 0 to n-1 bit positions are to be implemented.

It consists of an array of transistors, in which the number of rows equals the word length of data. The number of columns equals the maximum shift width. In this case, both are set. The control wires are routed diagonally through the array.

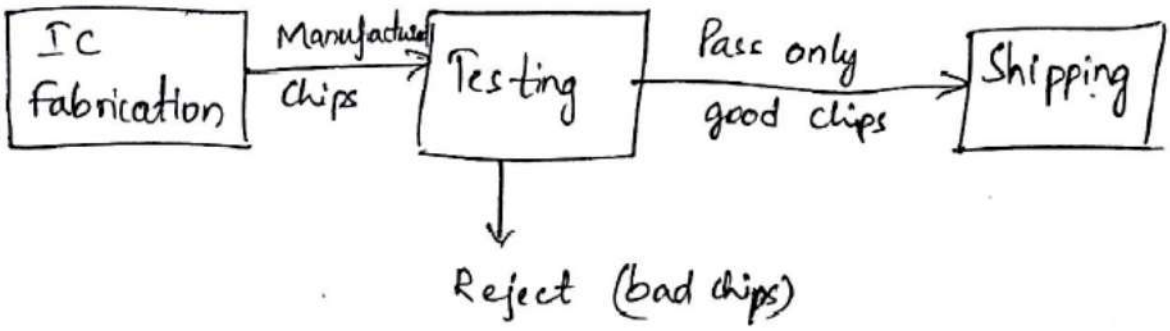
Regularity

Regularity: The hierarchical decomposition of a large system should result in not only simple, but also similar blocks, as much as possible. Regularity usually reduces the number of different modules that need to be designed and verified, at all levels of abstraction.

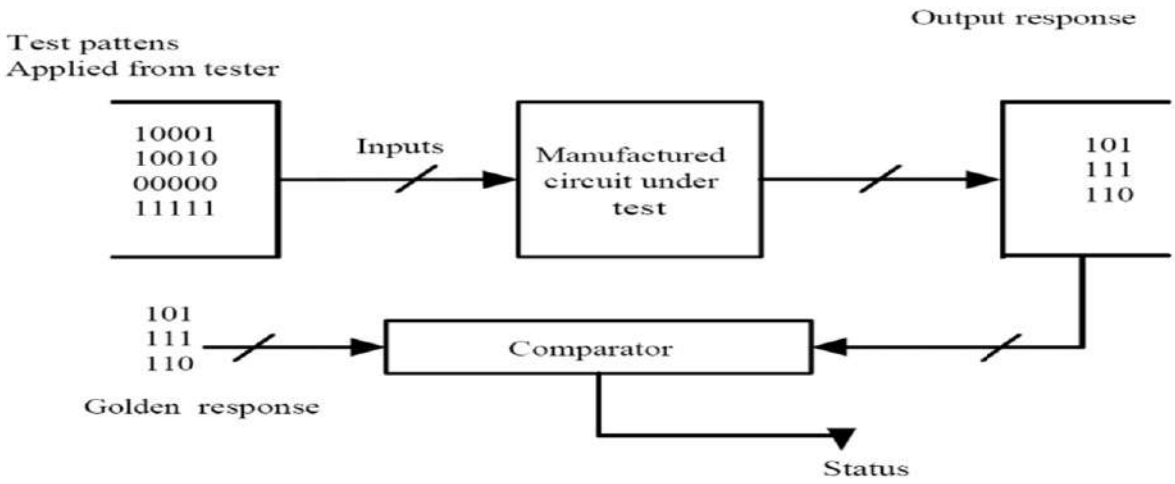


Practical Aspects and Testability

Introduction to Testing

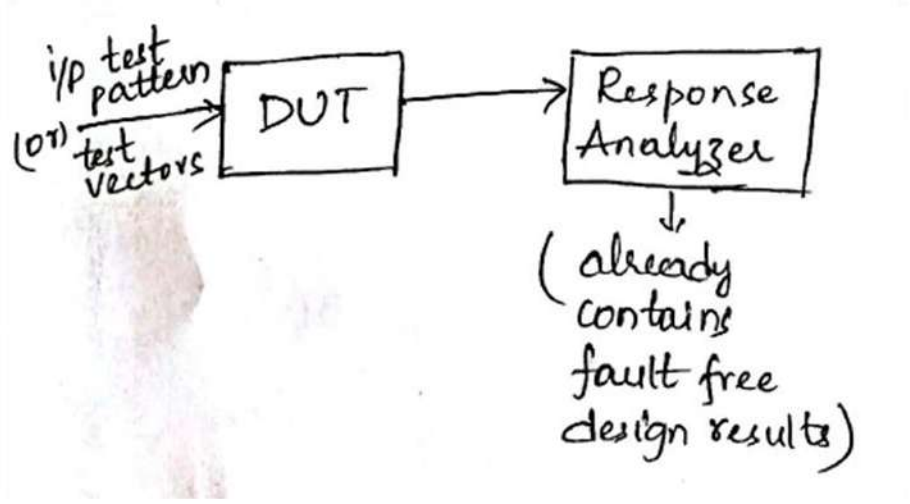


- Testing is a manufacturing step. It is required to guarantee a fault free product or it tells whether a system is good or bad.



How to test a chip

"Chip" while testing is called " **D**esign **U**nder **T**est" (DUT).



- Response analyzer contains **fault free design results stored in it**.

- When D.U.T produces the response corresponding to input test pattern, it compares this **response with already stored results**. If these are matched, D.U.T is fault free, else it has a fault and reject the chip.
- For generating input test pattern, we use a machine "ATPG".i.e "**A**utomatic **T**est **P**atten **G**enerator". The test pattern is a binary sequence.

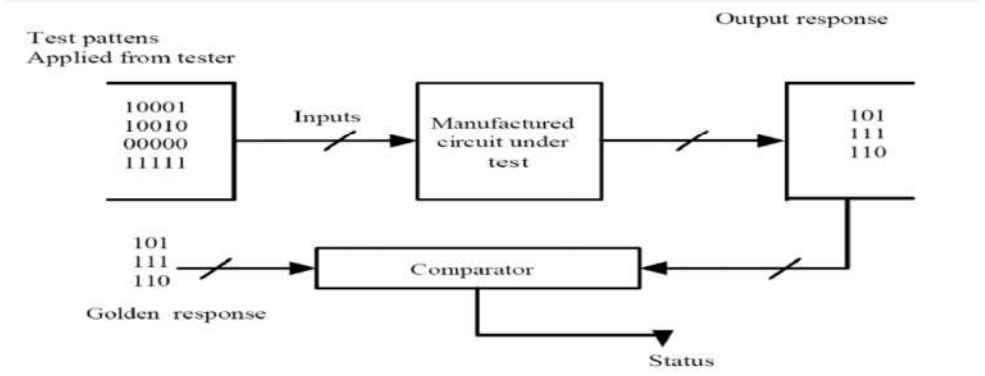
When to do Testing?

Can be carried out at various levels:

- At **chip level,** when chips are manufactured.
- At the **board level**, when chips are integrated on the board level.
- At the **system level,** when several boards are assembled together.

Test Generation

- A test is a sequence of test patterns, called **test vectors**, applied to the CUT whose outputs are monitored and analysed for the correct response.
- Exhaustive testing – applying all possible test patterns to CUT
- Functional testing – testing every truth table entry for a combinational logic CUT
  - Neither of these are practical for large CUTs
- **Fault coverage** is a quantitative measure of quality of a set of test vectors



Fault coverage of a given set of test vectors. Fault coverage=

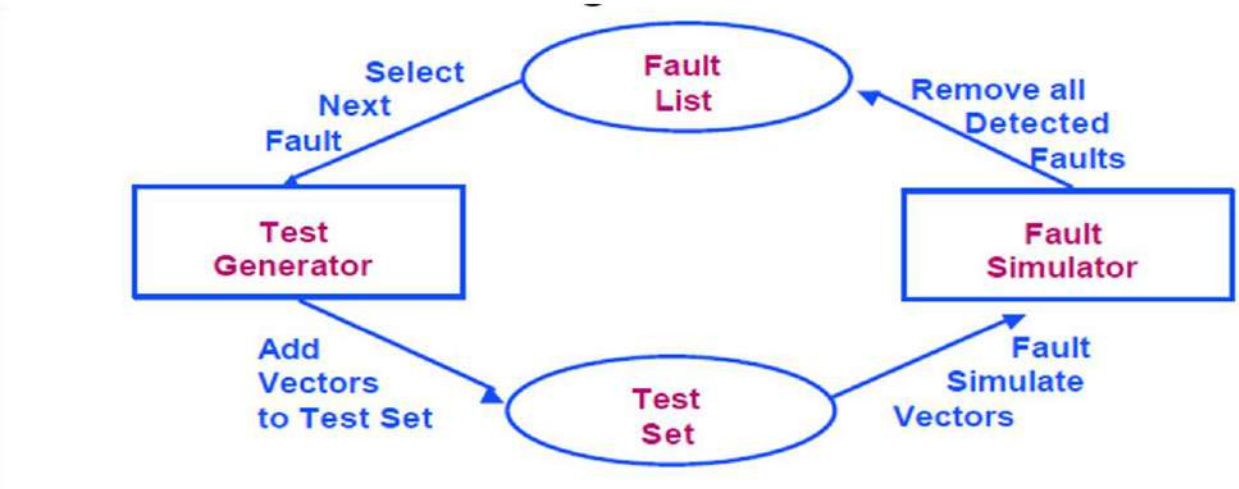
$$TC = \frac{\text{No. of faults detected}}{\text{No. of faults in the list}} \times 100$$

100% fault coverage may be impossible due to undetectable faults

$$\text{fault detection efficiency} = \frac{\text{number of detected faults}}{\text{total number of faults} - \text{number of undetectable faults}}$$

Test Generation System

Since ATPG is much slower than fault simulation, the fault list is trimmed with use o a fault simulator after each vector is generate.



Automatic Test Pattern Generation (ATPG)

- Algorithms generating sequence of test vectors for a given circuit based on specific fault models.

Fault Simulation

- Emulates fault models in CUT and applies test vectors to determine fault coverage.
- Simulation time can be reduced by parallel, deductive, and concurrent fault simulation.

Design For Testability (DFT)

- Design technique that makes test generation and test application easier and cost effective.
- It is very difficult to control and observe the internal flip flops.
- DFT techniques help in making the internal flip flops easily controllable and observable.
- Basically it converts the sequential circuit test generation problem to combinational circuit test generation problem.

Advantages of DFT:

- Shorter time to market
- Reduce design style
- Reduced cost
- To improve quality
- **Controllability:** setting or resetting nodes in the system by driving input pins of the chip.
- **Observability:** Observing a node in the system by watching external output pins of the chip.

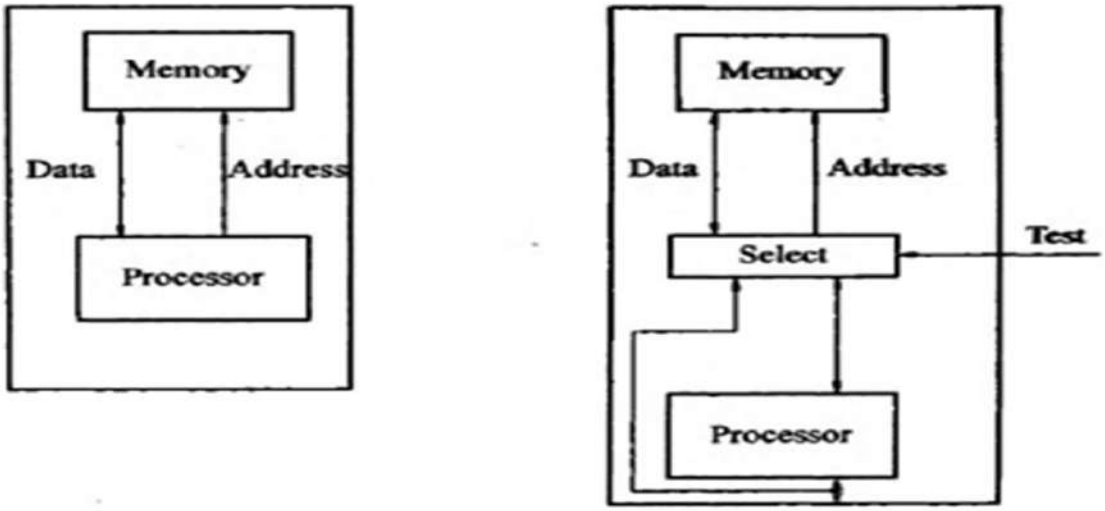
Disadvantages of DFT

- Chip area over head
- Performance overhead, i.e speed decreases

- DFT techniques generally fall into one of the following three categories:
  - ❖ Ad hoc DFT techniques
  - ❖ Level-sensitive scan design (LSSD) or scan design
  - ❖ Built-in self-test (BIST)

Ad hoc DFT technique

- Ad hoc testing combines a collection of tricks and techniques that can be used to increase the Observability and controllability of a design and that are generally applied in an application dependent fashion.
- It is a strategy to enhance the design testability without making much change to design style.



(a) Design with low testability (b) Adding a multiplexer (selector) improves testability.

- An example of such a technique is shows a simple processor with its data memory. Under normal configuration, the memory is only accessible through the processor.
- Writing and reading a data value into and out of a single memory position requires a number of clock cycles.
- The controllability and observability of the memory can be dramatically improved by add multiplexers on the data and address buses.
- During normal operation mode, these selectors direct the memory ports to the processor.
- During test, the data and address ports are connected directly to the I/O pins, and testing the memory can proceed more efficiently.
- It is often worth while to introduce extra hardware that has no functionality except improving the testability.

Advantage

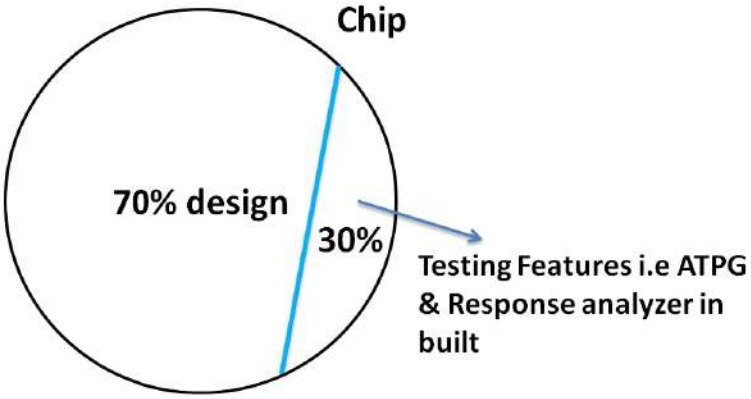
- It provides more systematic and automatic approach to enhance the design testability.

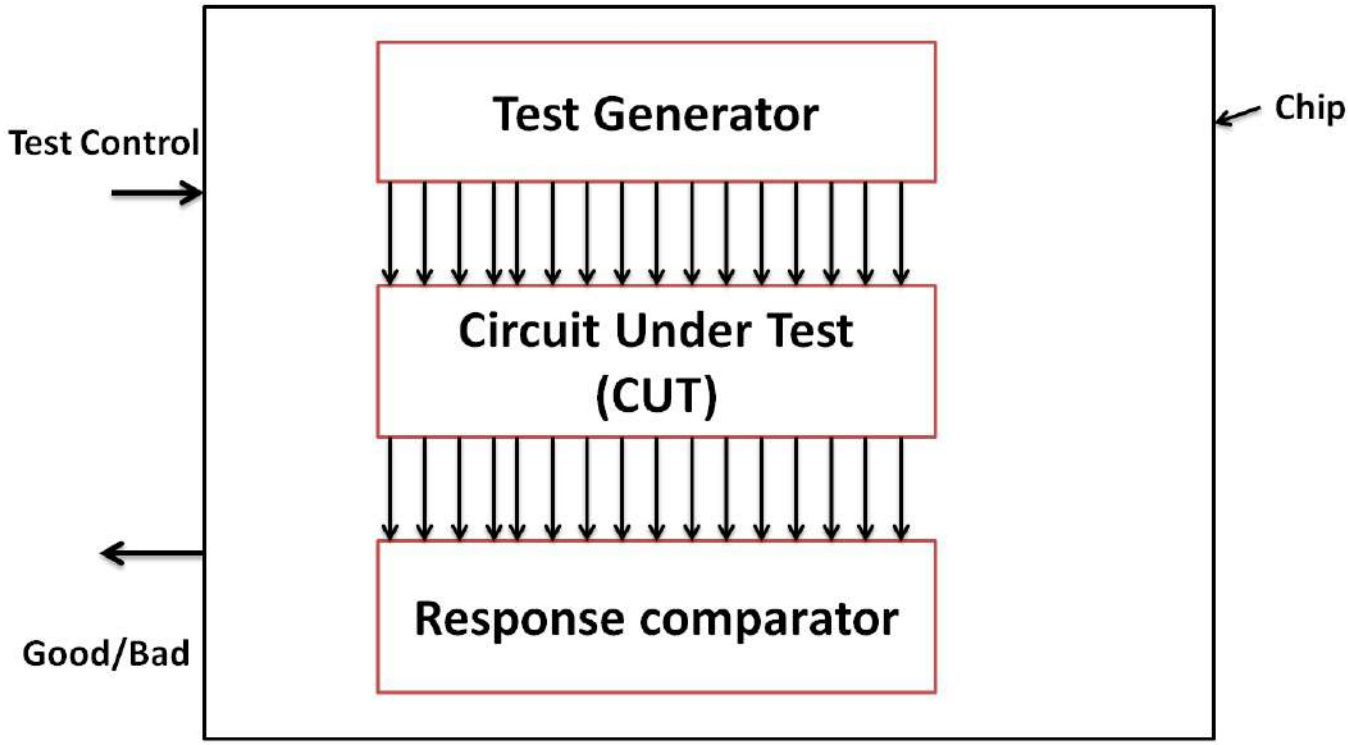
Disadvantages:

- Experts and tools **not** always available.
- Test generation must often be manually performed with no guarantee of high fault coverage.
- Design iterations may be needed, which is very time consuming.

BUILT IN SELF TEST (BIST)

- When circuits or design increase in size, testing them is a difficult process, i.e it increases the cost.
- So, we are including the testing features in the design itself, it simplifies or reduces our testing cost or procedure.
- Built-in self-test is the capability of a circuit (chip, board, or system) to test itself. BIST represents a merger of the concepts of built-in test (BIT) and self-test.



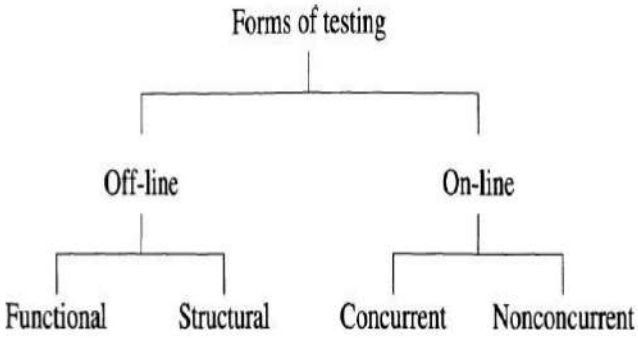


BIST techniques can be classified into two categories, namely

i. **On-line BIST**, which includes **concurrent and non-concurrent techniques**,
ii. **Off-line BIST**, which includes **functional and structural approaches**.

In on-line BIST, testing occurs during normal functional operating conditions; i.e., the circuit under test (CUT) is not placed into a test mode where normal functional operation is locked out. Concurrent on-line BIST is a form of testing that occurs simultaneously with normal functional operation. In non-concurrent on-line BIST, testing is carried out while a system is in an idle state. This is often accomplished by executing diagnostic software routines (macrocode) or diagnostic firmware routines (microcode). The test process can be interrupted at any time so that normal operation can resume.

Off-line BIST deals with testing a system when it is not carrying out its normal functions. Systems, boards, and chips can be tested in this mode. This form of testing is also applicable at the manufacturing, field, depot, and operational levels. Often Off-line testing is carried out using on-chip or on-board test- pattern generators (TPGs) and output response analyzers (ORAs). Off-line testing does not detect errors in real time, i.e., when they first occur, as is possible with many on-line concurrent BIST techniques.



- ✓ **Functional off-line BIST** deals with the execution of a test based on a functional description of the CUT and often employs a functional, or high-level, fault model.
- ✓ **Structural off-line BIST** deals with the execution of a test based on the structure of the CUT.

Off-Line BIST Architectures

Off-line BIST architectures at the chip and board level can be classified according to the following criteria:

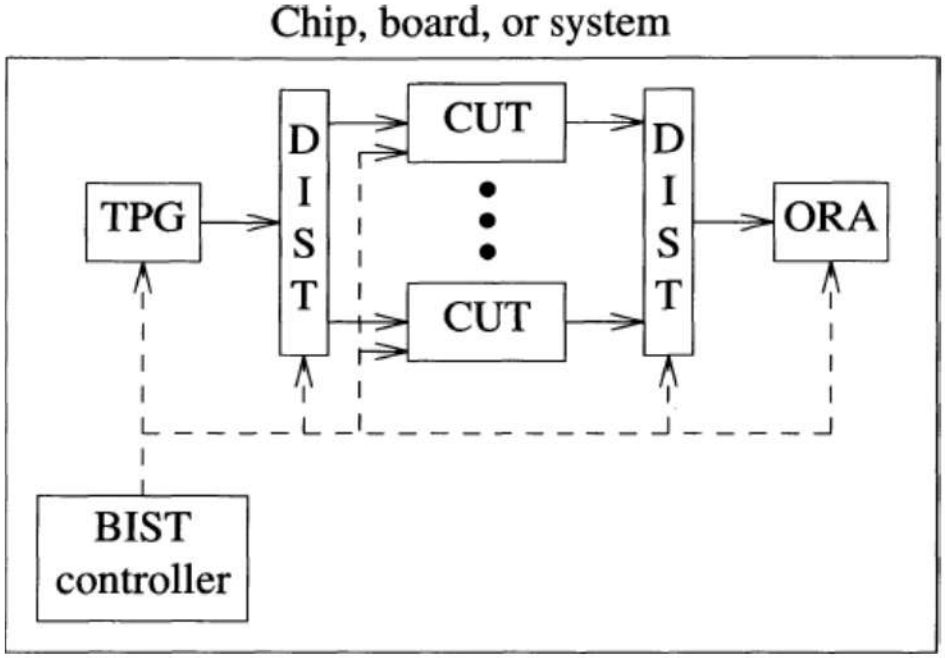
1. Centralized or distributed BIST circuitry;
2. Embedded or separate BIST elements.

BIST architectures consist of several key elements, namely

1. Test-pattern generators;
2. Output-response analyzers;
3. The circuit under test;
4. A distribution system (DIST) for transmitting data from TPGs to CUTs and from CUTs to ORAs;
5. BIST controller for controlling the BIST circuitry and CUT during self-test.

Centralized BIST architecture

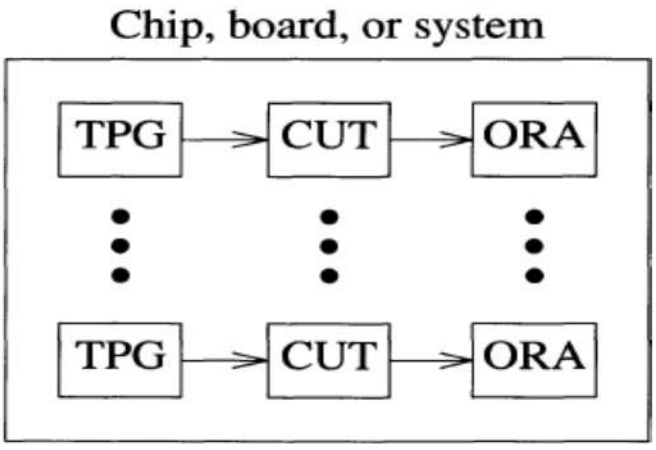
The general form of a centralized BIST architecture is shown in the below figure.



Here several CUTs share TPG and ORA circuitry. This leads to reduced overhead but increased test time. During testing, the BIST controller may carry out one or more of the following functions:

1. Single-step the CUTs through some test sequence.
2. Inhibit system clocks and control test clocks.
3. Communicate with other test controllers, possibly using test busses.

Distributed BIST architecture



The distributed BIST architecture is shown in above figure. Here each CUT is associated with its own TPG and ORA circuitry. This leads to more overhead but less test time and usually more accurate diagnosis

Advantages

- Test patterns generated on chip, controllability increased.
- Test can be on line or off line
- Test can run at circuit speed, more realistic; shorter test time; easier delay testing
- External test equipment greatly simplified
- Easily adapting to engineering changes.

Ten Marks Questions

1. Explain the two phase clocking with circuit diagram in sequential circuits.
2. Draw the 4 bit dynamic shift register using NMOS and CMOS with stick diagrams
3. Explain bus line architecture for NMOS and CMOS with different classes.
4. Discuss the general arrangement of a 4-bit arithmetic process.
5. Explain the design of a 4-bit shifter in detail.
6. Explain in detail about design for testability.
7. What is design for testability? Discuss about the built in self test.