2019 7b 2014 7c 2018 8. a) describe logical system design

Q. What do you mean by system design? Describe logical and physical design.

Ans: <u>System design:</u> The design phase is a transition from a user – oriented document to a document oriented to the programmers or data base personnel.

<u>Logical design:</u> When analysts prepare the logical system design, they specify the user needs at a level of detail that virtually determines the information flow into and out of the system and the required data resources. The design covers the following:

1. Review the current physical system – its data flows, files content, volumes, frequencies etc.
2. Prepares output specifications – that is determines the format, content, and frequency of reports, including terminal specifications and locations.
3. Prepares input specifications – format, content and most of the input functions.
4. Prepares edit, security and control specifications.
5. Specify the implementation plan.
6. Prepares a logical design walkthrough of the information flow, output, input, controls and implementation.
7. Reviews benefits, costs, target dates and system constraints.

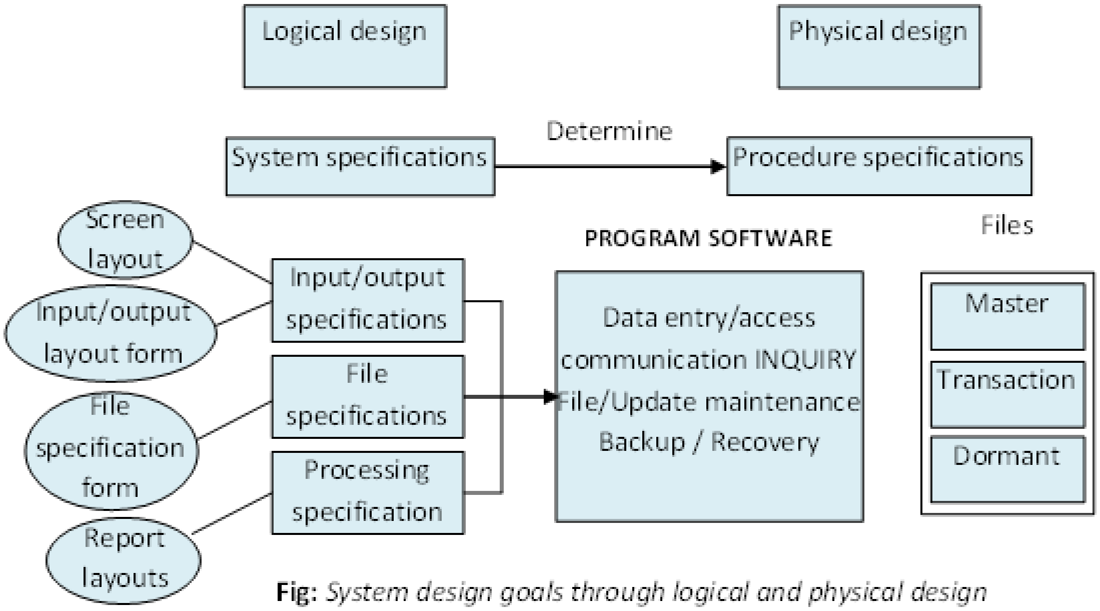

Fig: *System design goals through logical and physical design*

<u>Physical design:</u> Physical system design consists of the following steps:

1. Design the physical system.
    1. Specify input/output media.
    2. Design the data base and specify backup procedures.
    3. Design physical information flow through the system and a physical design walkt6hrough.
2. Plan system implementation.
    1. Prepare a conversion schedule and a target date.
    2. Determining training procedure, courses and timetable.
3. Devise a test and implementation plan and specify any new hardware/software.
4. Update benefits, costs, conversion date, and system constraints.

Q. Mention some design methodologies and explain any one of them.

Ans: There are three methodologies –

1. Structured design

2019 7a) Discuss the concept of structured design 2.75

2. Forms driven methodology
3. Structured walkthrough

Structured design: Structured design is a data – flow – based methodology. The approach begins with a system specification that identifies inputs and outputs and describes the functional aspects of the system.
Structured design partitions a program into small, independent modules. They are arranged in a hierarchy that approximates a model of the business area ad is organized in a top – down manner with the details shown at the bottom. Thus, structured design is an attempt to minimize complexity and make a problem manageable by subdividing it into smaller segments, which is called modularization or decomposition. In this way, structuring minimizes intuitive reasoning and promotes maintainable, provable systems.

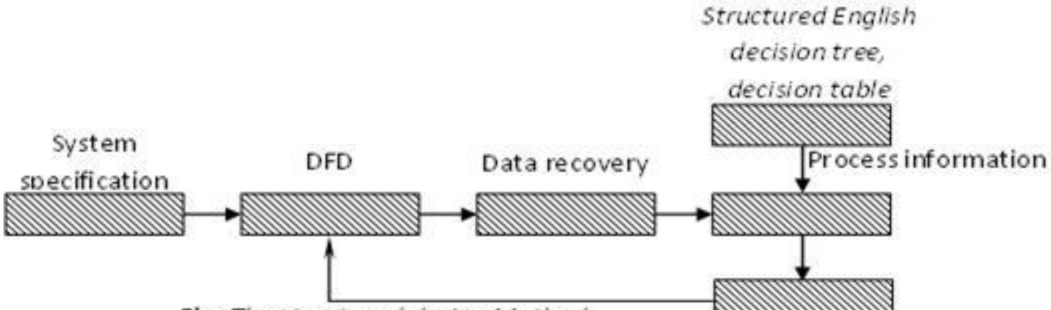

Fig: *The structured design Method*

A design is said to be top – down if it consists of a hierarchy of modules, with each module having a single entry and a single exit subroutine. The primary advantages of this design are as follows:

1. Critical interfaces are tested first.
2. Early versions of the design, through incomplete are useful enough to resemble the real system.
3. Structuring the design, per se, provides control and improves morale.
4. The procedural characteristics define the order that determines processing.

Q. Describe functional decomposition? Or, define module, connection and couple.
Ans: Functional decomposition: The documentation tool for structured design is the hierarchy or structure chart. It is a graphic tool for representing hierarchy and it has three elements:

1. **Module:** The module is represented by a rectangle with a name. it is a contiguous set of statements.

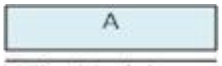

Fig: *Module*

1. **Connection:** The connection is represented by a vector linking two modules. It usually means one module has called another module. In the figure, module A calls module B; it also calls module C.

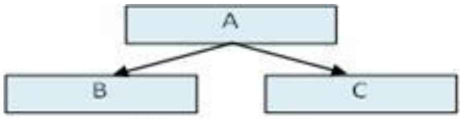

Fig: *Connection*

2019 7c coupling, modular coupling and modular cohesion 3

1. **Couple:** The couple is represented by an arrow with a circular tail. It represents data items moved from one module to another. In figure O, P, and Q are couples. Modules A calls B, passing o downward. Likewise, module A calls C, passing P downward and receiving Q back.

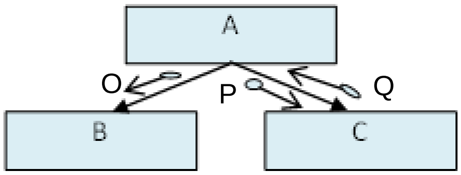

Fig: *Coupling*

Q. Define module coupling and module cohesion.

Ans: Module coupling: Module coupling refers to the number of connections between a "calling" and a "called" module and the complexity of these connections. There must be at least one connection between a module and a calling module. A design objective for producing an easily understood code is to make the modules as independent as possible.

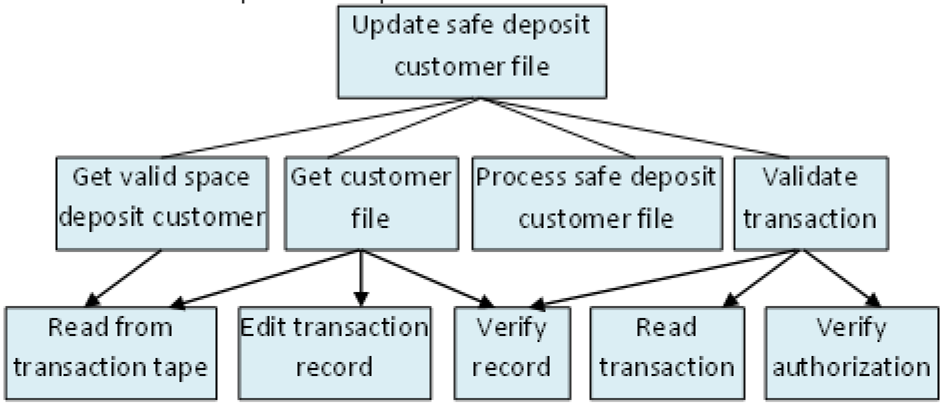
Update safe deposit customer file
Get valid space deposit customer
Get customer file
Process safe deposit customer file
Validate transaction
Read from transaction tape
Edit transaction record
Verify record
Read transaction
Verify authorization

Fig: *Module cohesive design*

bond

Module cohesion: Module cohesion refers to the relationship among elements within a module. If a module does more than one discrete task, the instructions in that module are said not to be bound together very closely. Modules that perform only one task are said to be more cohesive than modules that perform multiple task.

Q. Write short notes on HIPO. Hierarchical Input / Processing / Output

Ans: HIPO: HIPO is a forms – driven technique in that standard forms are used to document the information. It consists of a hierarchy chart and an associated set of input/process/output charts. HIPO captures the essence of top down decomposition; it describes the data input and output from processes and defines the data flow composition. It was developed by IBM as a design aid and implementation technique with the following objectives:

1. Provide a structure by which the functions of a system can be understood.
2. State the functions to be performed by the program rather than specifying the program statements to be used to perform the functions.
3. Provide a visual description of input to be used and output to be produced for each level of the diagram. HIPO makes the transformation of input to output data visible.

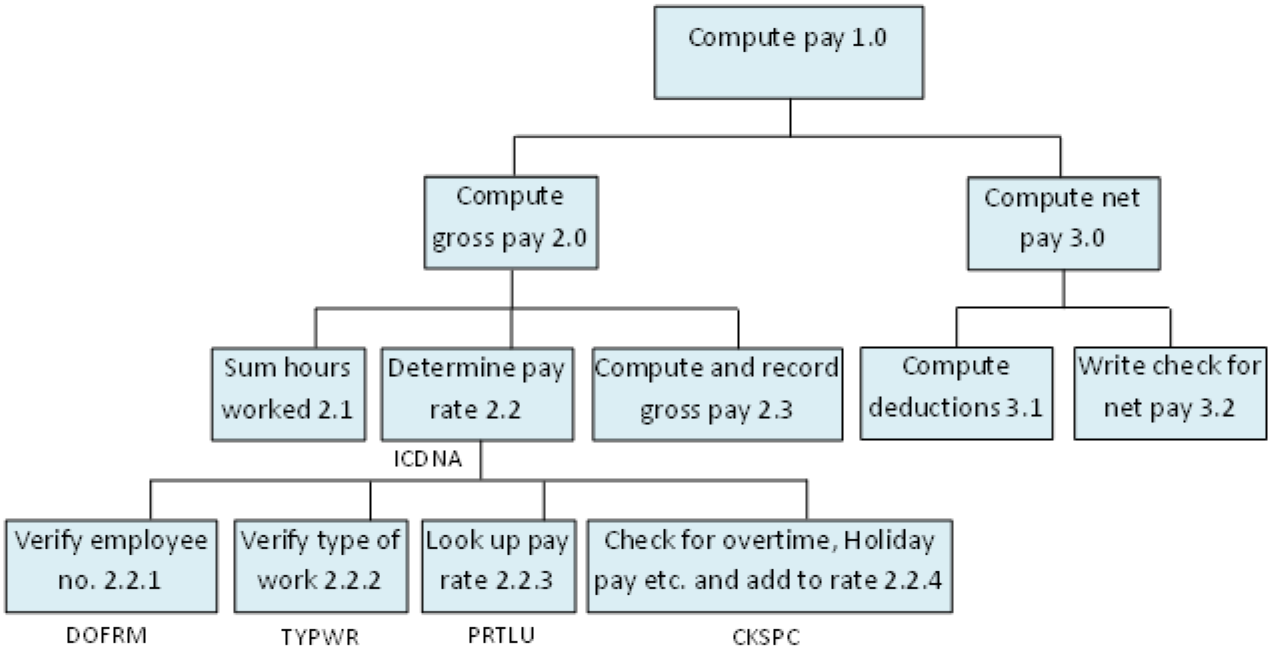
Compute pay 1.0
Compute gross pay 2.0
Compute net pay 3.0
Sum hours worked 2.1
Determine pay rate 2.2
Compute and record gross pay 2.3
Compute deductions 3.1
Write check for net pay 3.2
ICDNA
Verify employee no. 2.2.1
Verify type of work 2.2.2
Look up pay rate 2.2.3
Check for overtime, Holiday pay etc. and add to rate 2.2.4
DOFRM
TYPWR
PRTLU
CKSPC

Fig: *A hierarchy diagram.*

The procedure for generating HIPO diagrams is simple:

1. Begin at the highest level of abstraction and define the inputs to the system and the outputs from it in aggregate terms.
2. Identify the processing steps by those that convert input into output.
3. Document each element using HIPO diagram notation and the associated tree like structure.
4. Identify sub process and their respective inputs and outputs. Continue decomposition until the processes cannot be decomposed any further.

The HIPO package format consists of the following:

1. Visual table of contents shows the structure of the diagram and the relationships of the functions in a hierarchical manner.
2. Overview diagrams describe the major functions and reference the detail diagram needed to expand the functions adequately.
    1. The input section that contains the data items used by the process steps.
    2. The output section that contains the data items created by the process steps.
    3. Process section that contains numbered steps that describes the functions to be performed.
    4. The extended description refers to non – HIPO documentation and code.
3. Detail diagram contains an extended description section that amplifies the process steps and references the code associated with each process step.

Q. What is the need for HIPO?

Ans: It is important to use HIPO early in the design phase of a project so that designers can document their thoughts concurrently with the design process. Thus, the preparation of HIPO diagrams is a by – product of the thought process of the design rather than an additional chore.