General consideration;

Static data flow analysis is a design technique developed by M.A.jackson which focuses on processing data flow instead of control flow. According to Jackson, the program structure should reflect the structure of data it processes. There are four step in Static data flow analysis which are as follows:

1. Evaluate the problem environment and define the structures for data to be processed.

There are four basic components for defining data which are as follows:

1. Elementary operation: parts which cannot be further subdivided and take on values.
2. Sequence: Two or more parts occurring one time in a specific sequence.
3. Iteration: A part which occurs zero, one or more times.
4. Selection: two or more parts in which one, and only one occurs.
5. Develop program structure based on the data structures.
6. Define the task to be performed basically non-procedural in terms of elementary operations to suitable components of the program structure.
7. Allocate each of the elementary operations to suitable components of the program structures.

In general, application of traditional design techniques by N individuals of the same problem leads to N solutions or at least a large number of related but different solutions. But Application of Jackson’s method tends to produce the same design by all individuals or at least a limited number of closely related designs.

The analysis metrics used for static data flow analysis are:

* Coupling
* Miller’s Law
* Graciunas’s Law
* Factoring
* Black Boxes
* Scope of Control/Effect

Coupling:

This system shows very high level of control coupling and low level of data coupling between the modules. Each module can be implemented independently, and does not need to know how the other modules are implemented.

Miller’s Law:

The Design doesn’t exhibit any modules that exceeds 7 ± 2 interactions. We can clearly see from the figure that there are not any modules which handles more than 4 modules at a time which basically tells us that it doesn’t violates the miller’s law. Thus this design technique will reduces complexity, implementation, modification and maintenance cost.

Graciunas’s Law: The system is highly factored and the majority of data is passed from one module to another, using a one-way connection where each module only communicates with its direct superior controlling module.

Factoring: Our design technique is highly factored. Each module contains only one idea, and the upper modules are involved in control and the lower level modules are implementation details. Being highly factored, it ensures that each modules complies with Miller’s Law, Graciunas’s Law scope of control and scope of control as well.

Black-Boxes: if we examine the diagram, we can see that the system exhibits the characteristics of black –boxes. The modules used by the system doesn’t depend on the construction of these how these modules were constructed and module does exactly the same function for every module that invokes it.

Scope of control/effect: The system already defines the module which is highly factored. So, it is guaranteed to have ideal scope. The high level module only effects the sub-module below it and it doesn’t affect any module above it. If we examine the diagram, we can clearly see that the scope of effects are subset of scope of control which is very good quality for a design.