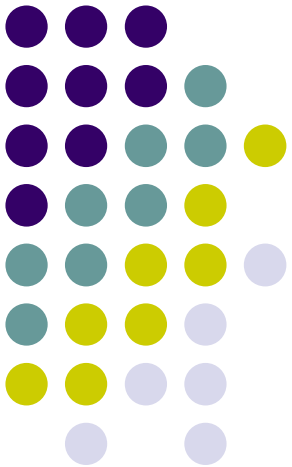
SEMICON Solutions

IP Design Manual

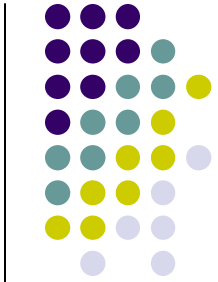


Created: Duong Dang

Date: 10/03/09

SEMICON
THINKING OUT OF THE BOX.

Content

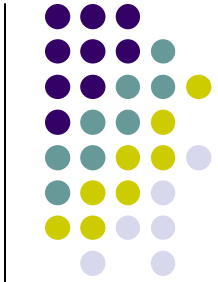


- Design Procedure
- Verification Plan Procedure
- Implementation Process

Content

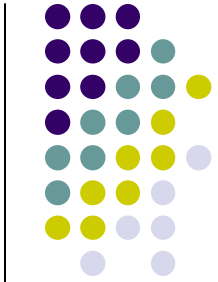
- **Design Procedure**
- Verification Plan Procedure
- Implementation Process

Design Procedure



- Basic Policies
- Create the development plan
- Design function specification
- Design module detail

Basic Policies



- The design procedure is recommened when IP is designed
- Design according to this procedure
- However, this procedure deviates according to the planning of development and requirement for design module.
- In this case, new procedure must be clear and output items and the quality equal with the procedure showed here

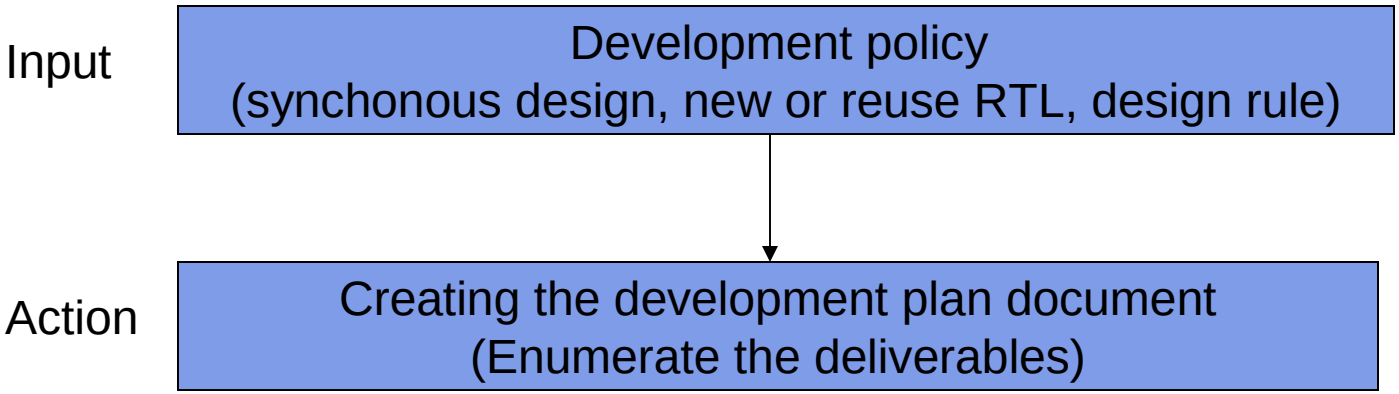
Create development plan

- Do this when IP is developed and designed

Input	Development policy (synchonous design, new or reuse RTL, design rule)

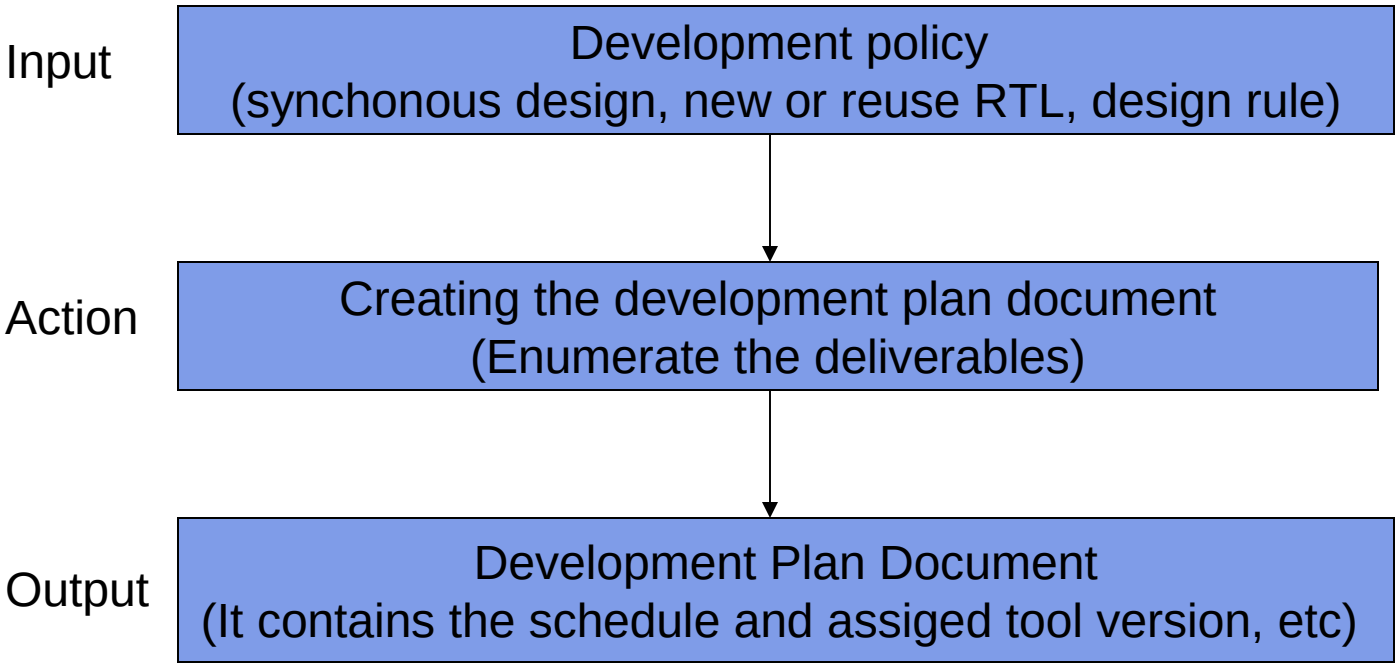
Create development plan

- Do this when IP is developed and designed



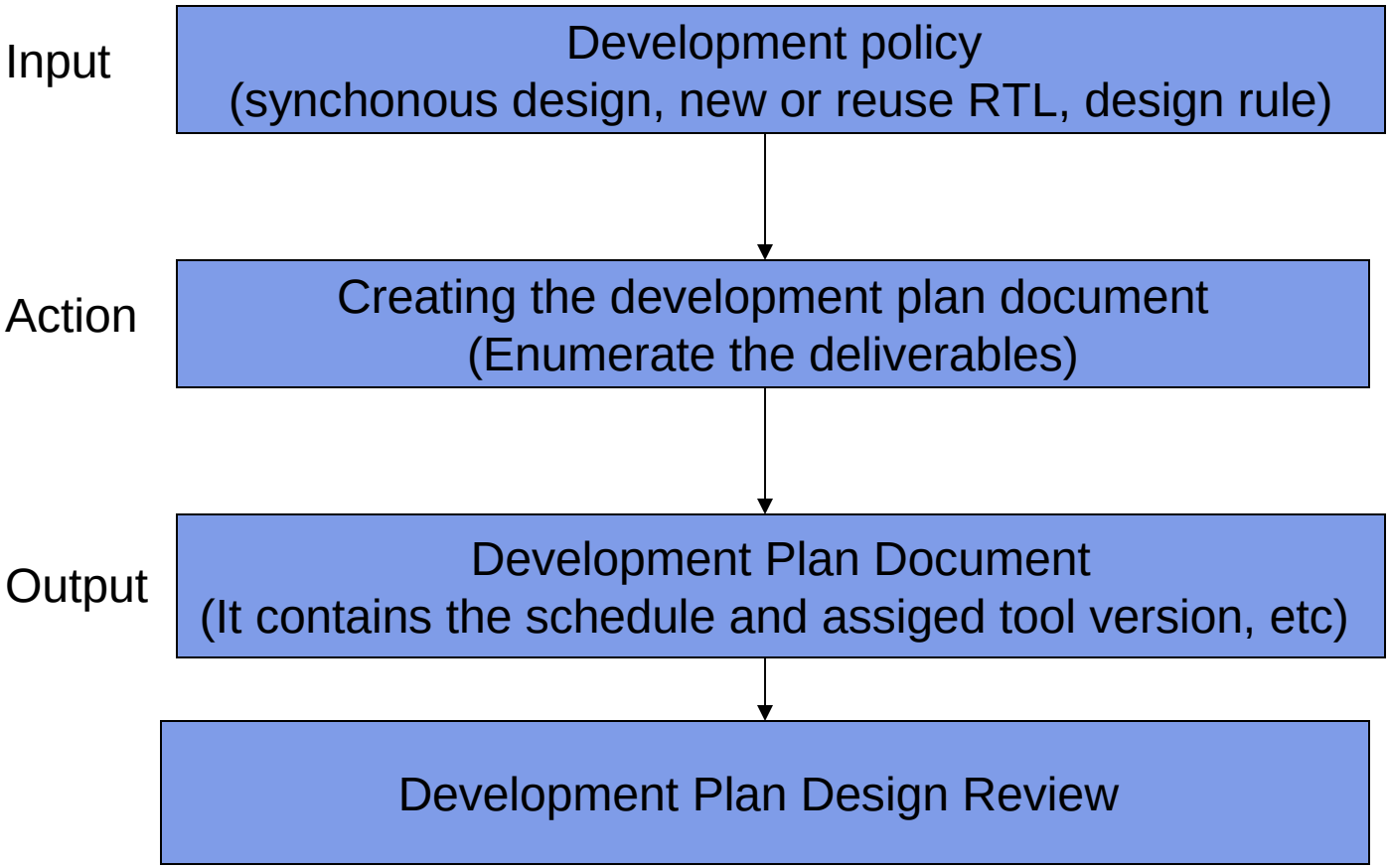
Create development plan

- Do this when IP is developed and designed



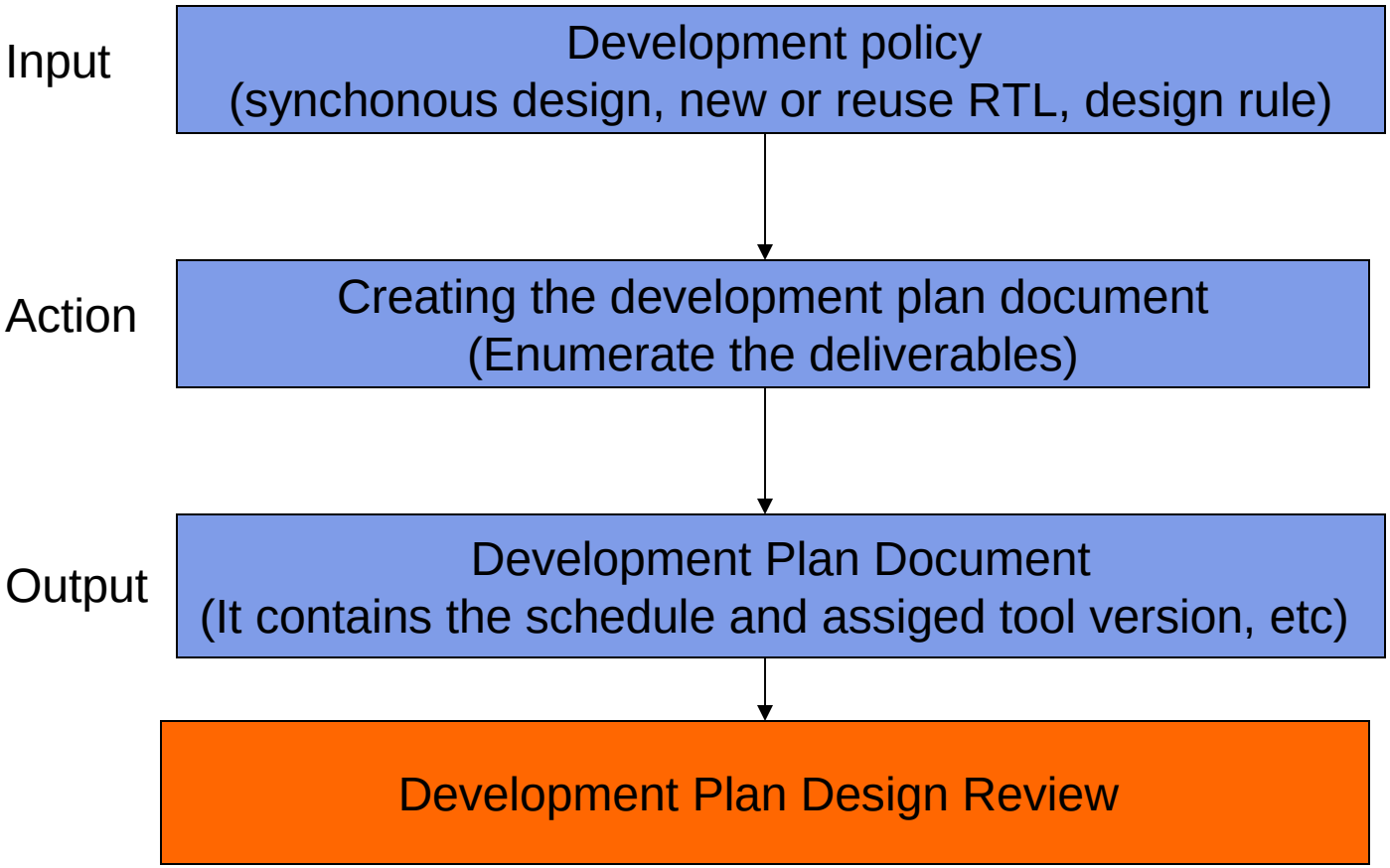
Create development plan

- Do this when IP is developed and designed

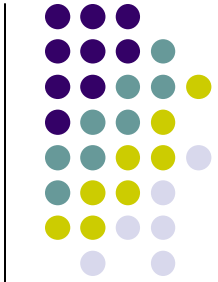


Create development plan

- Do this when IP is developed and designed

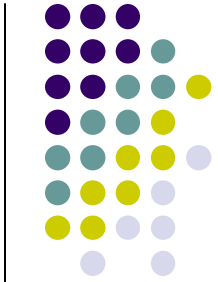


Design the function spec



- Function specification design defines functions and operations of desiged module clearly
- Function spec must be neither lack or contradiction
- Description should be easy to understand

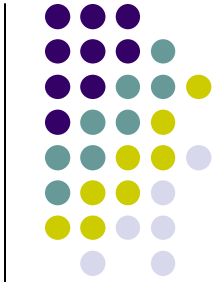
Design the function spec

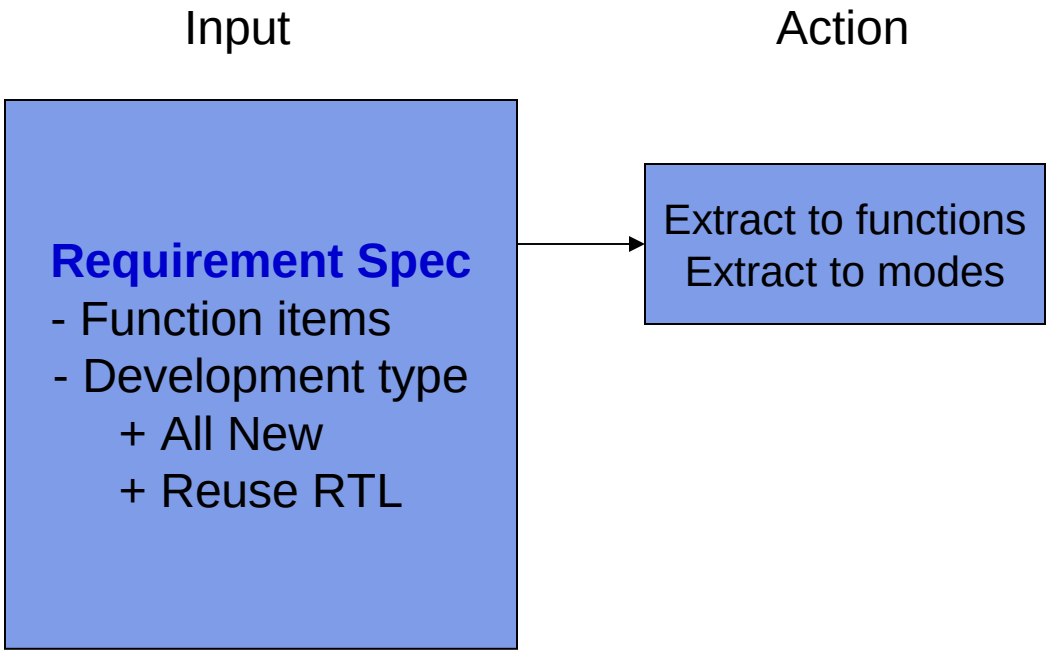


Input

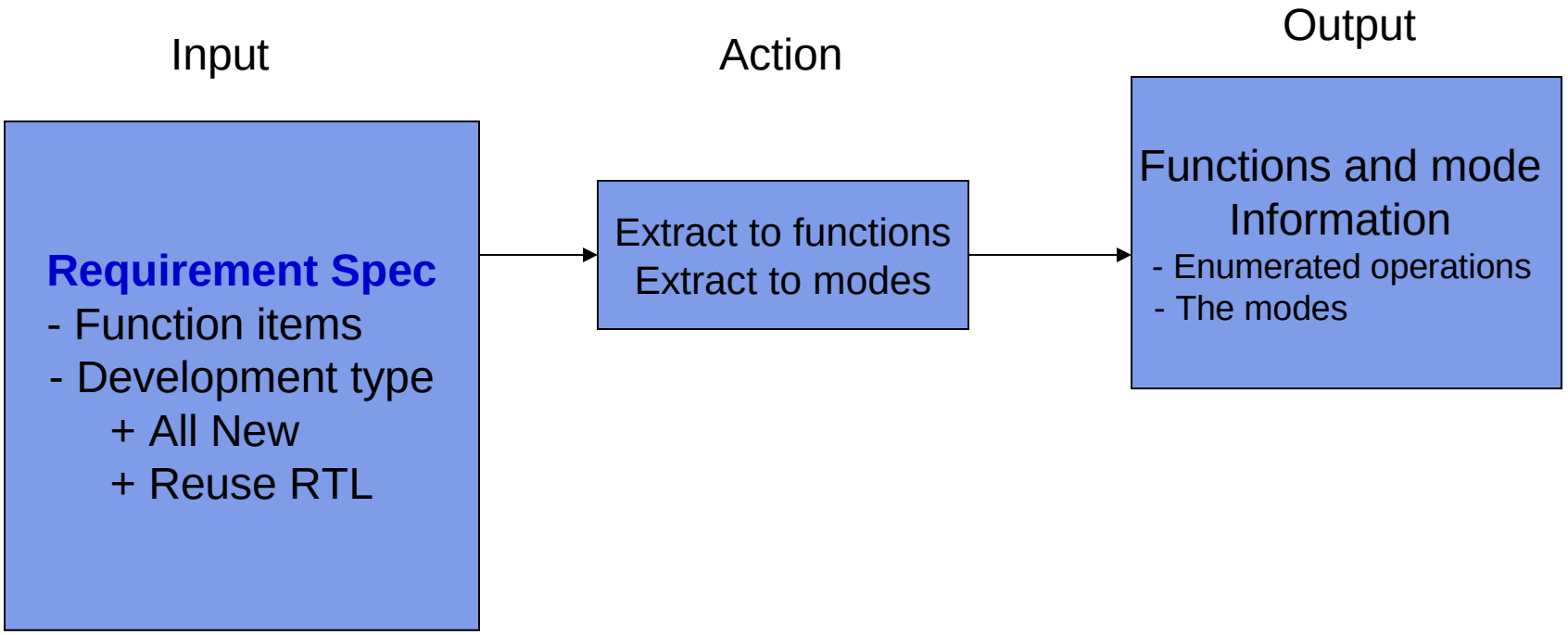
Requirement Spec
- Function items
- Development type
  - + All New
  - + Reuse RTL

Design the function spec

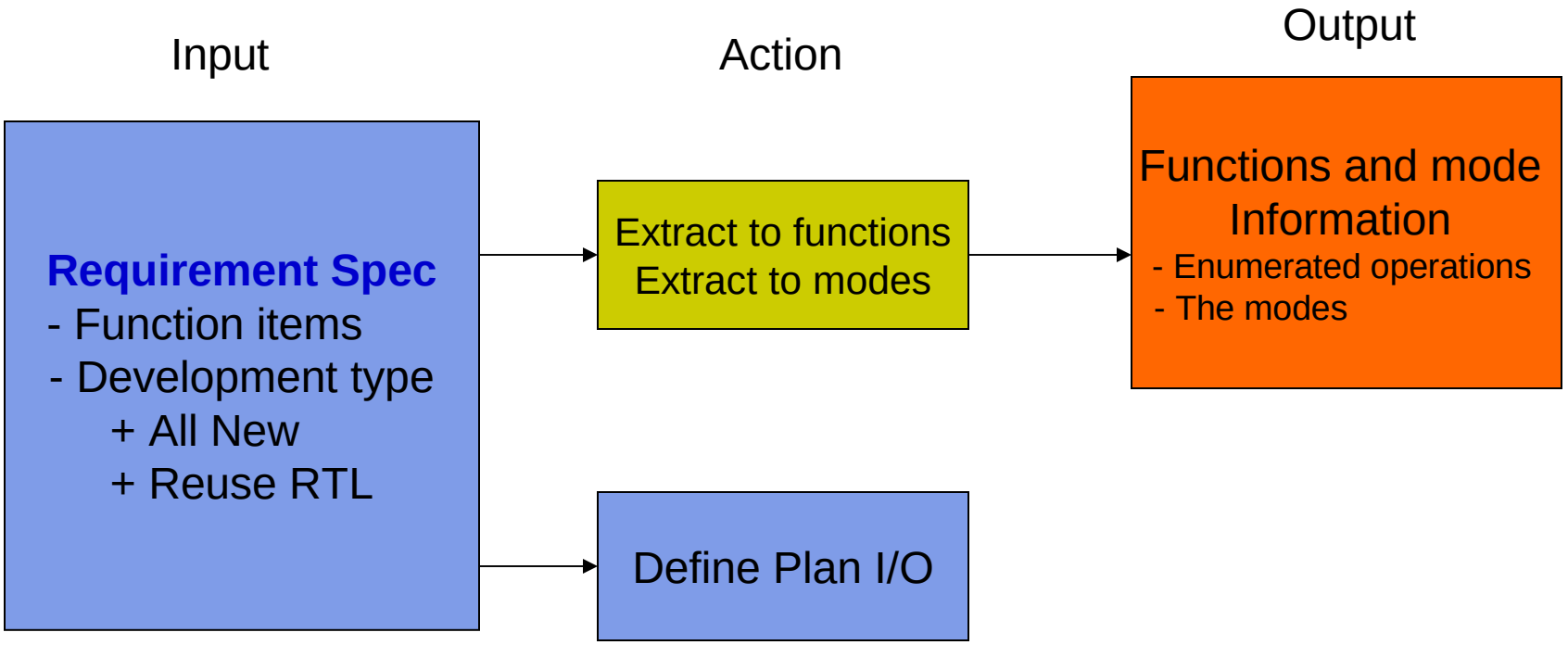




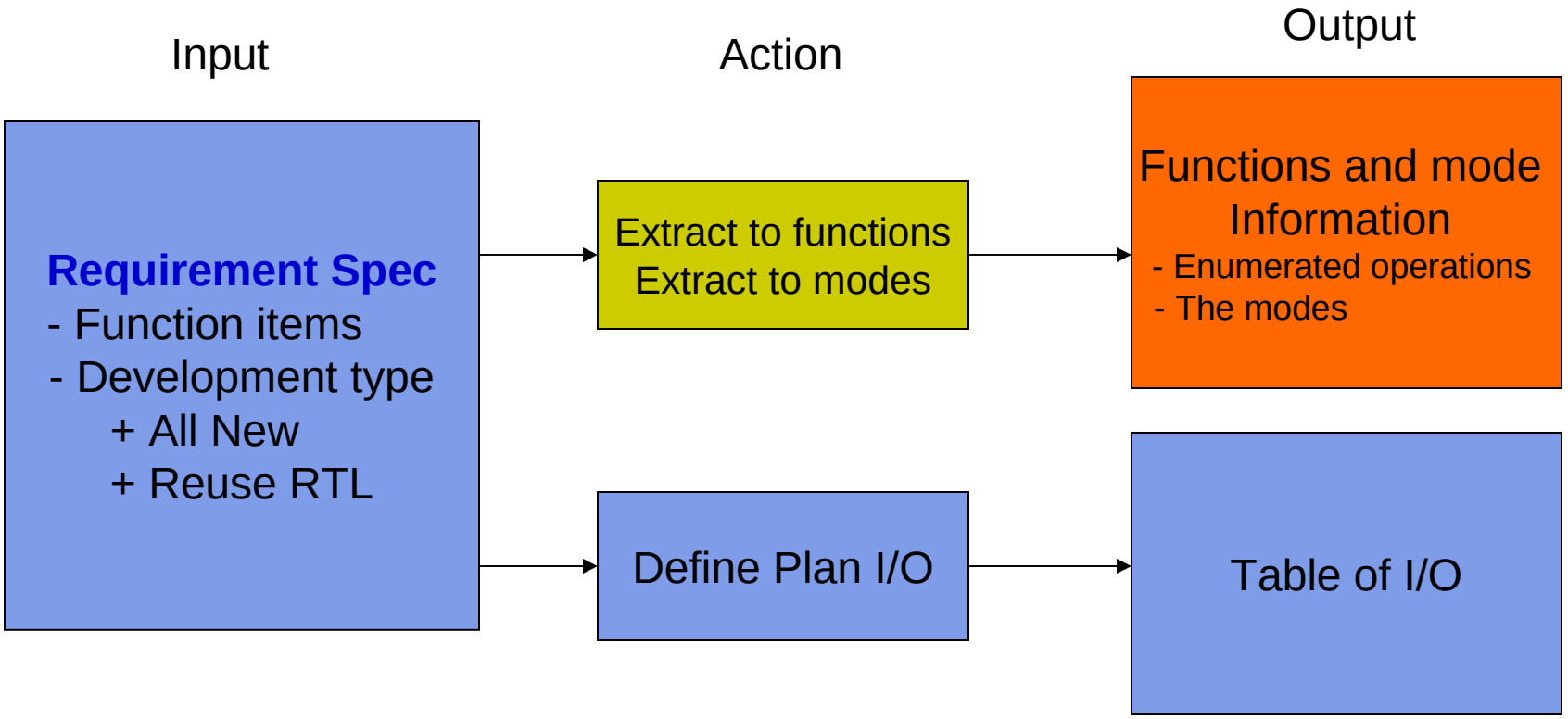
Design the function spec



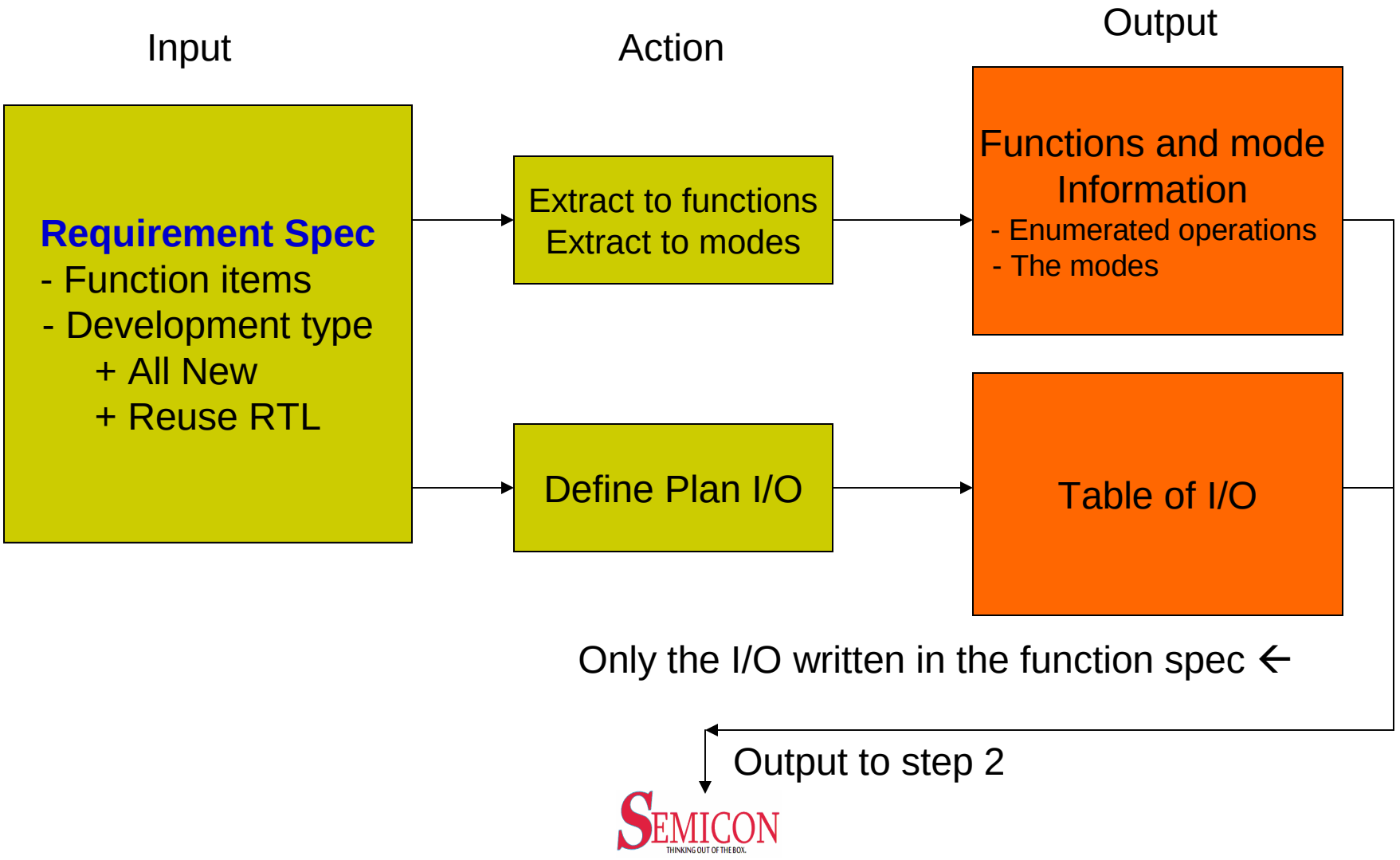
Design the function spec



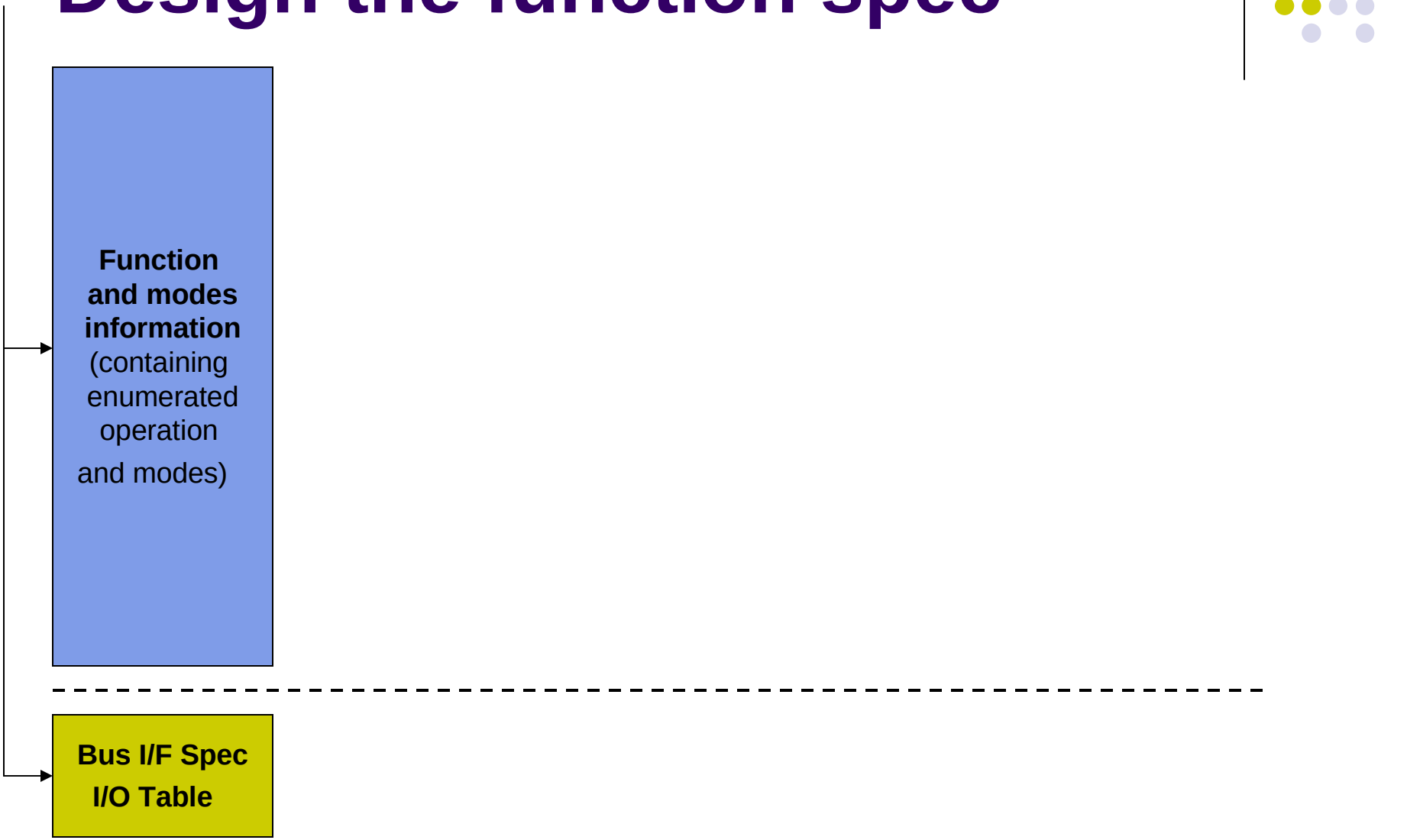
Design the function spec



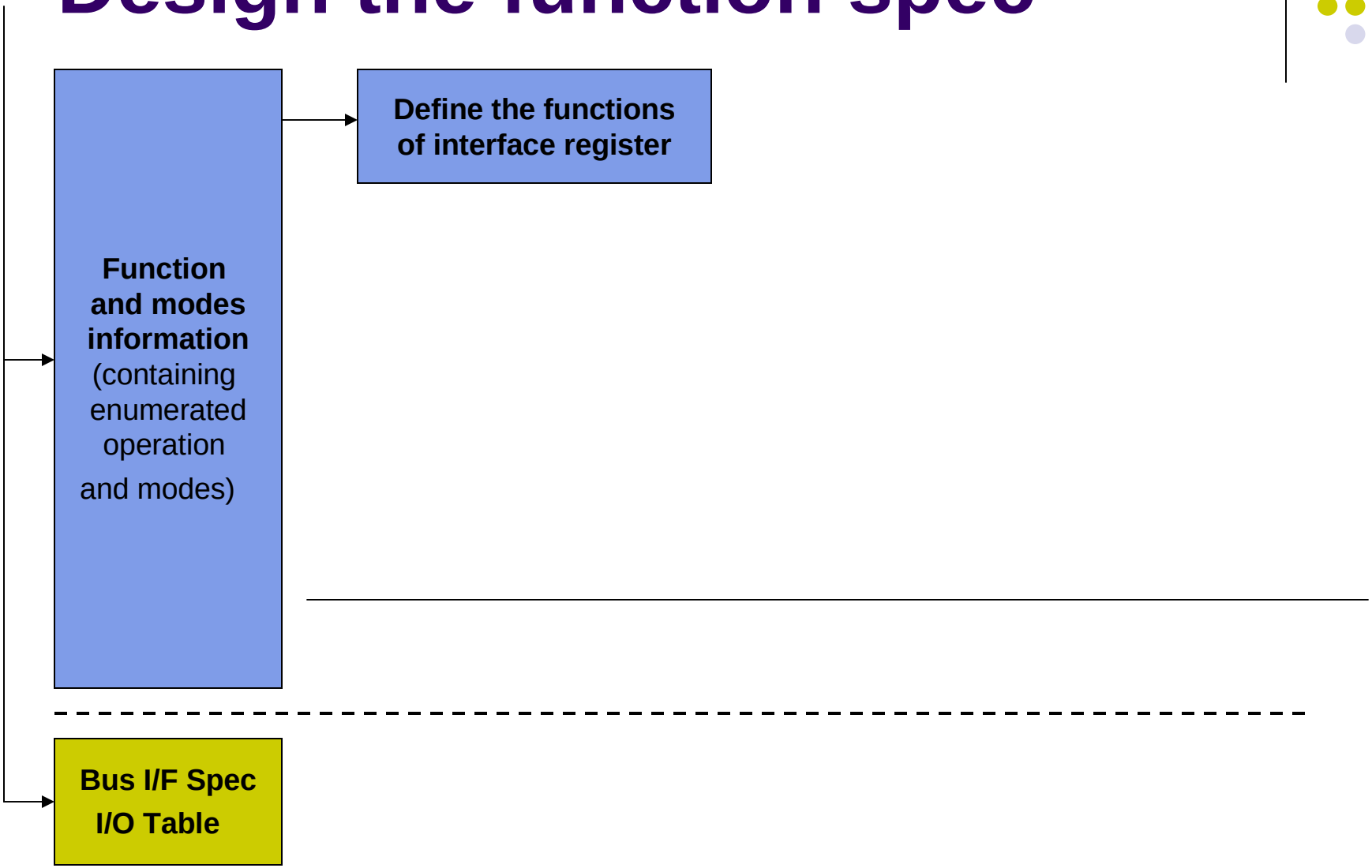
Design the function spec



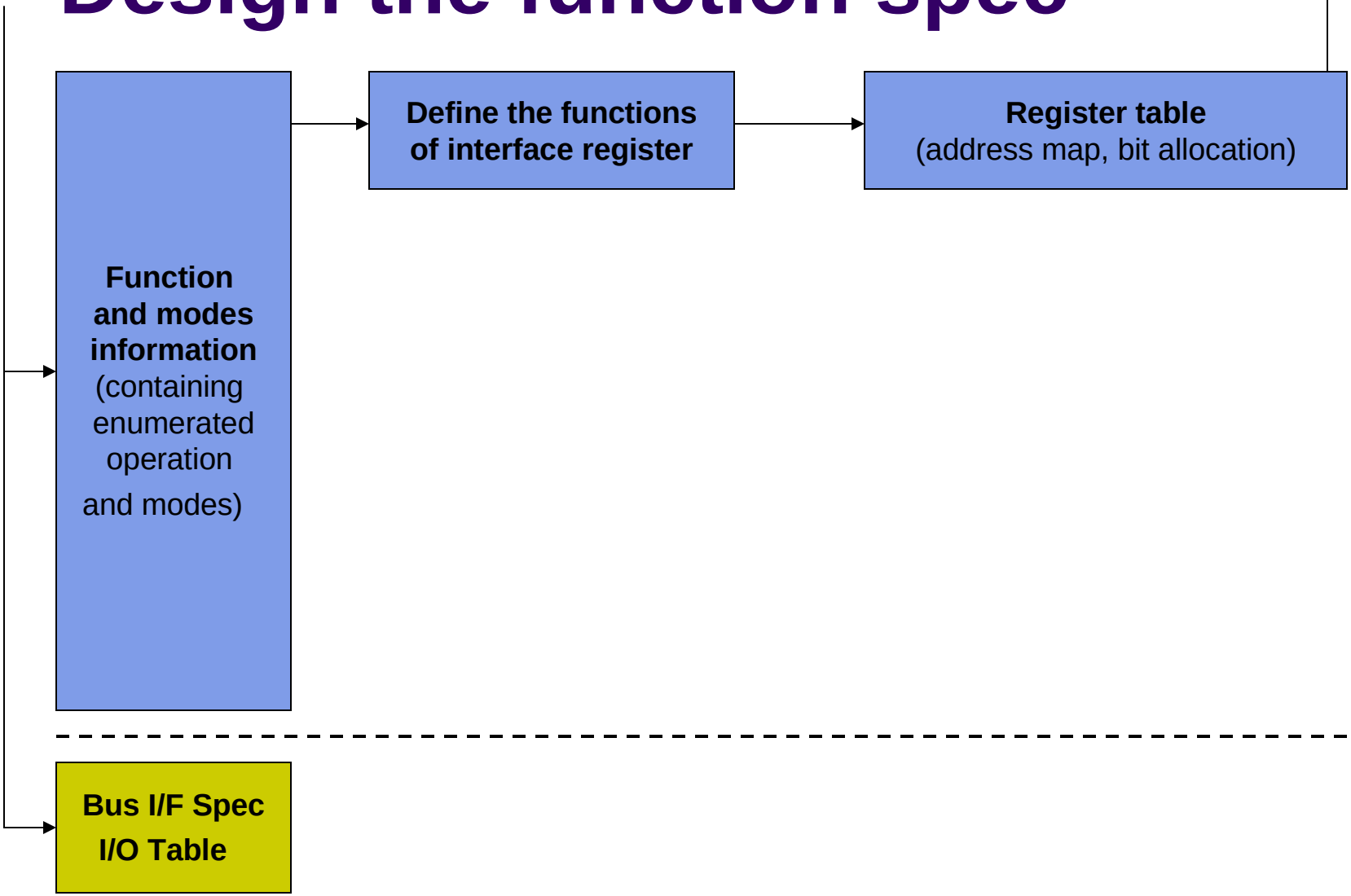
Design the function spec



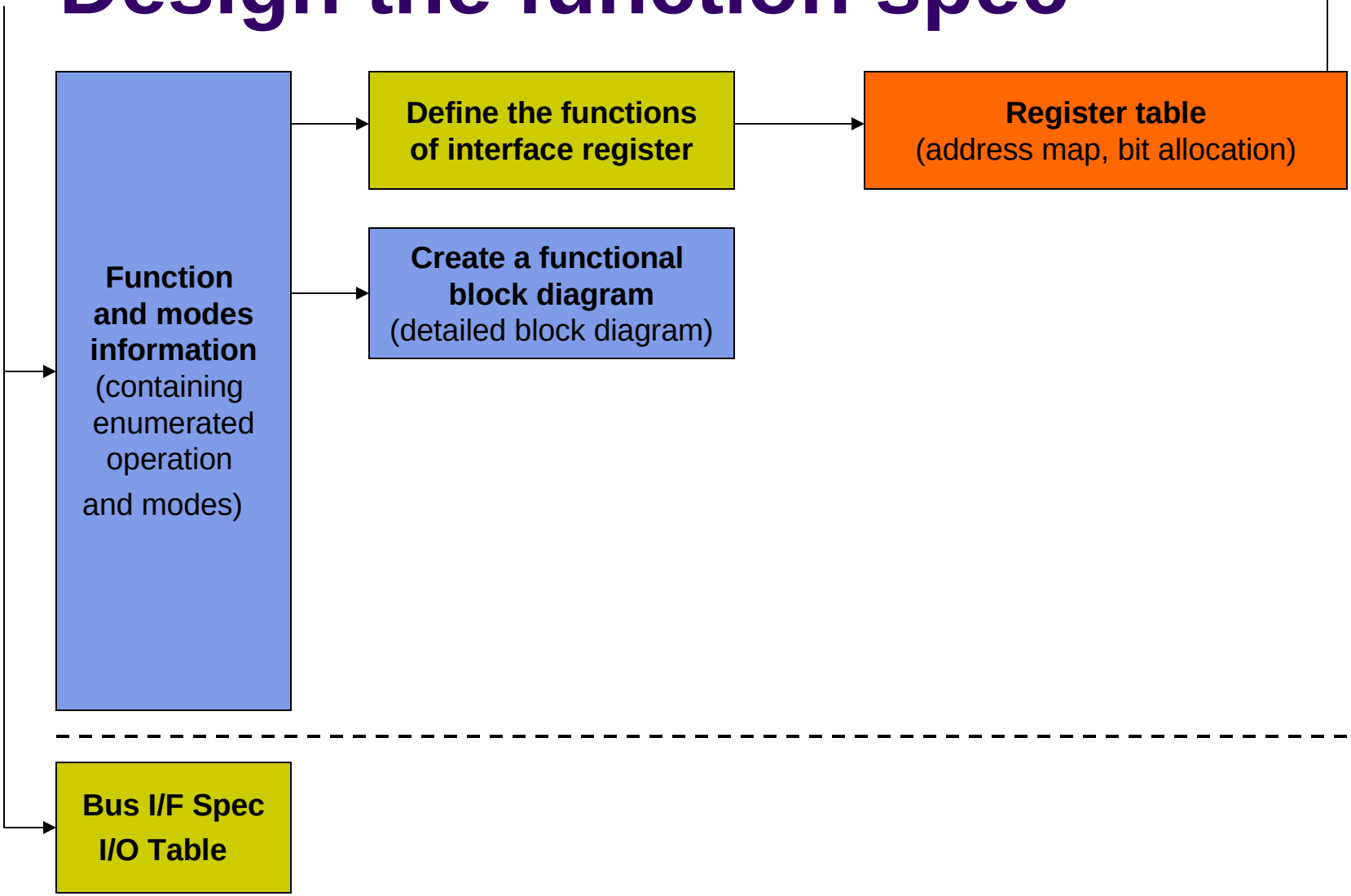
Design the function spec



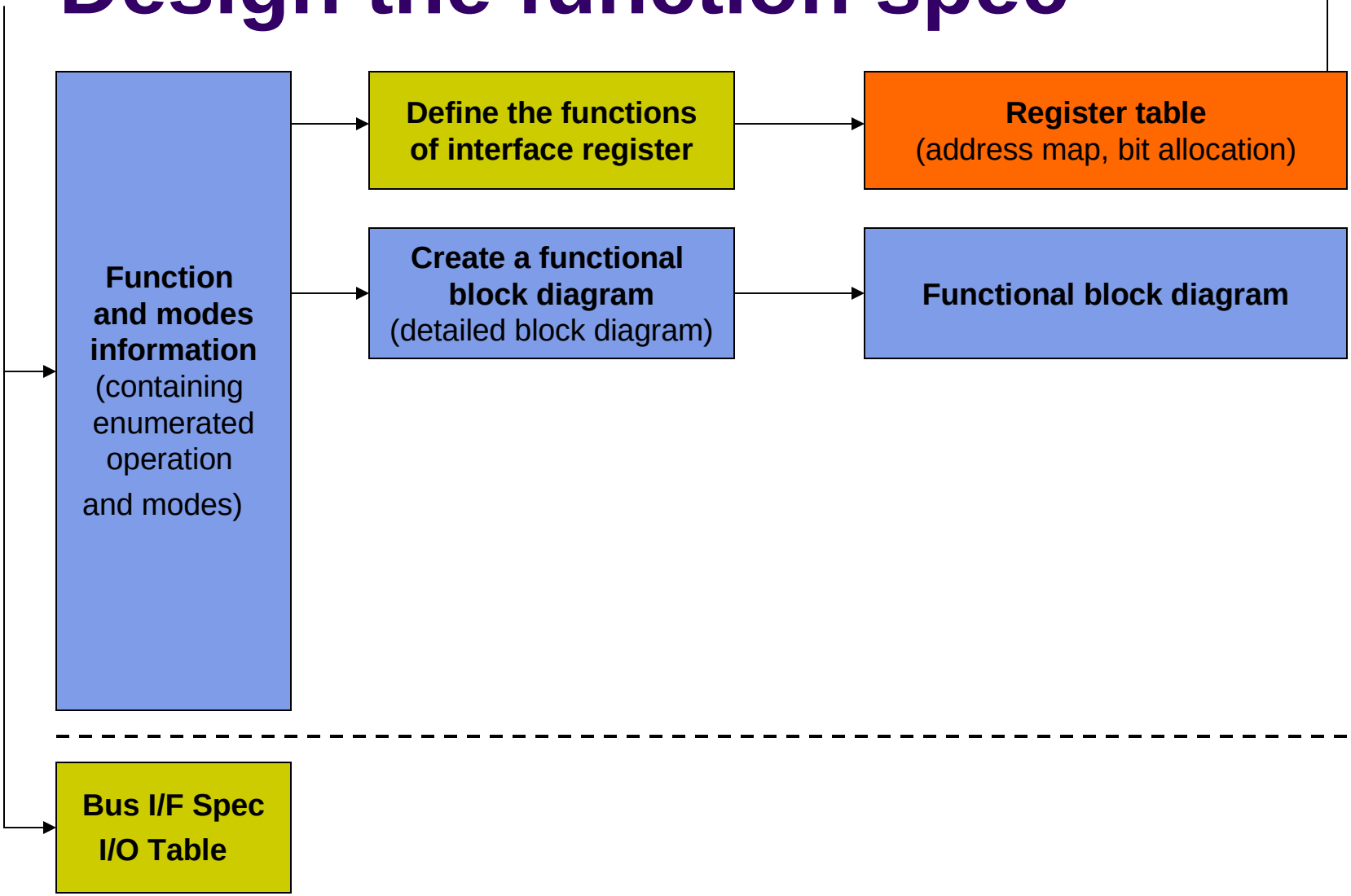
Design the function spec



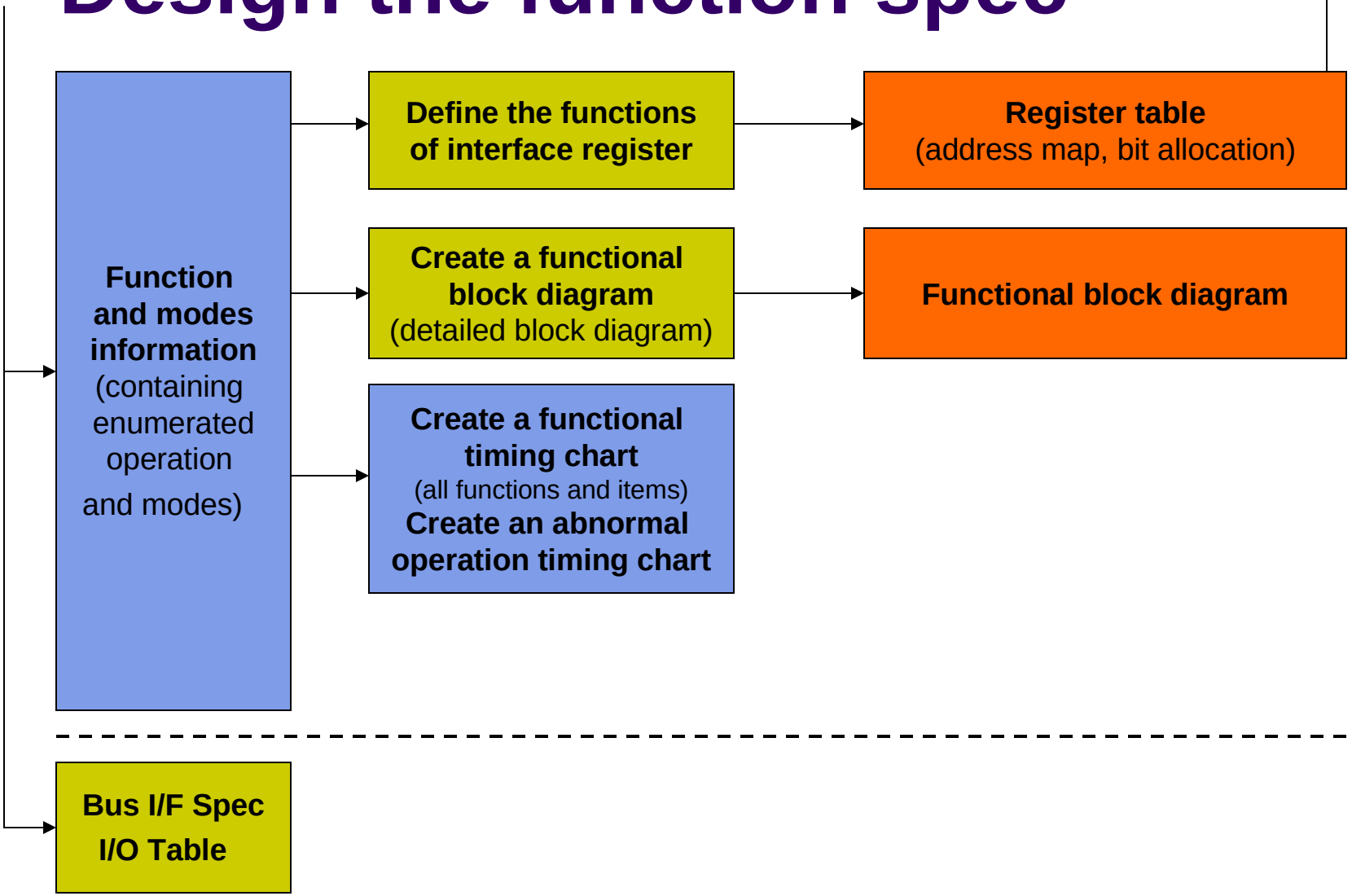
Design the function spec



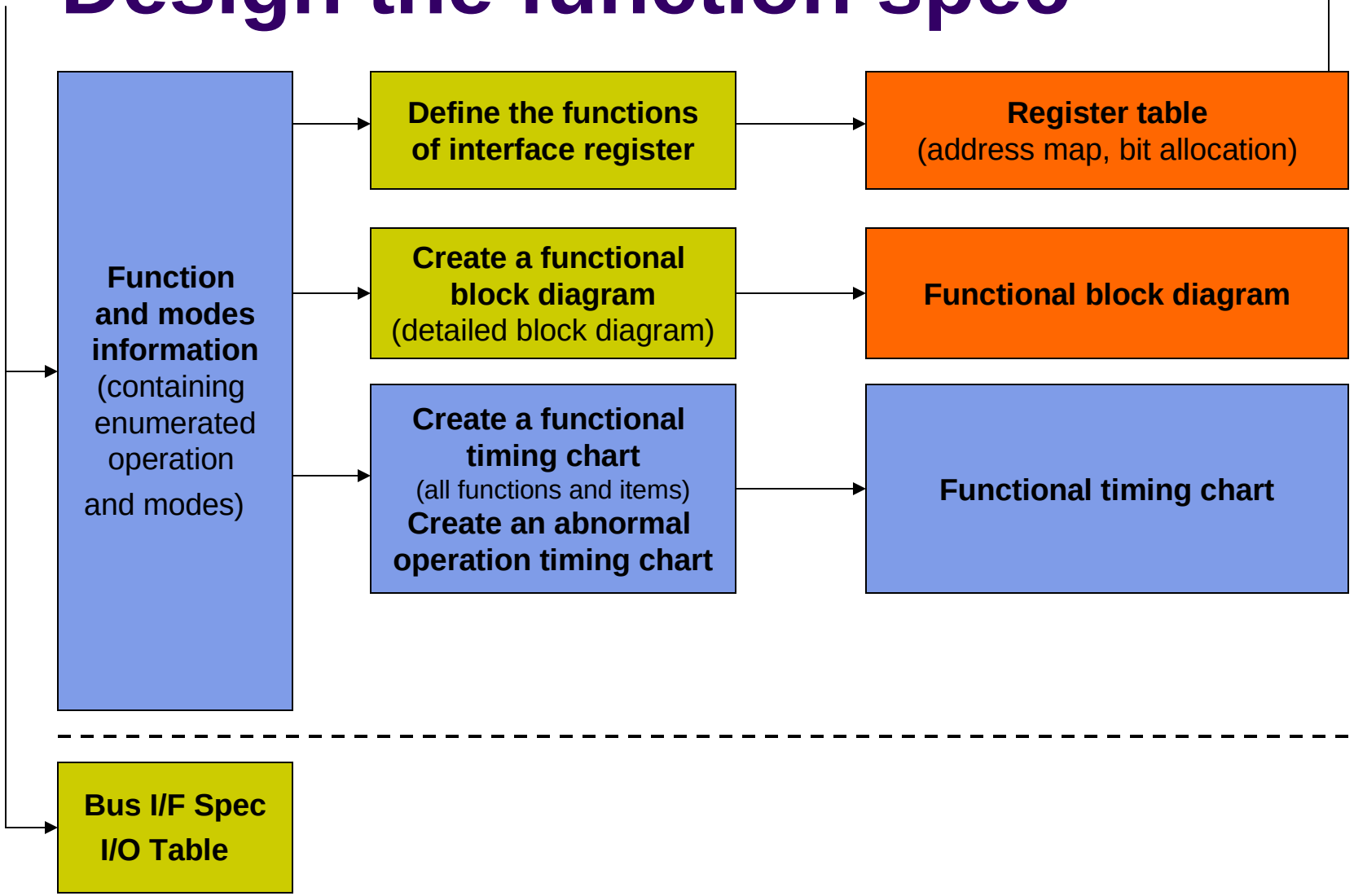
Design the function spec



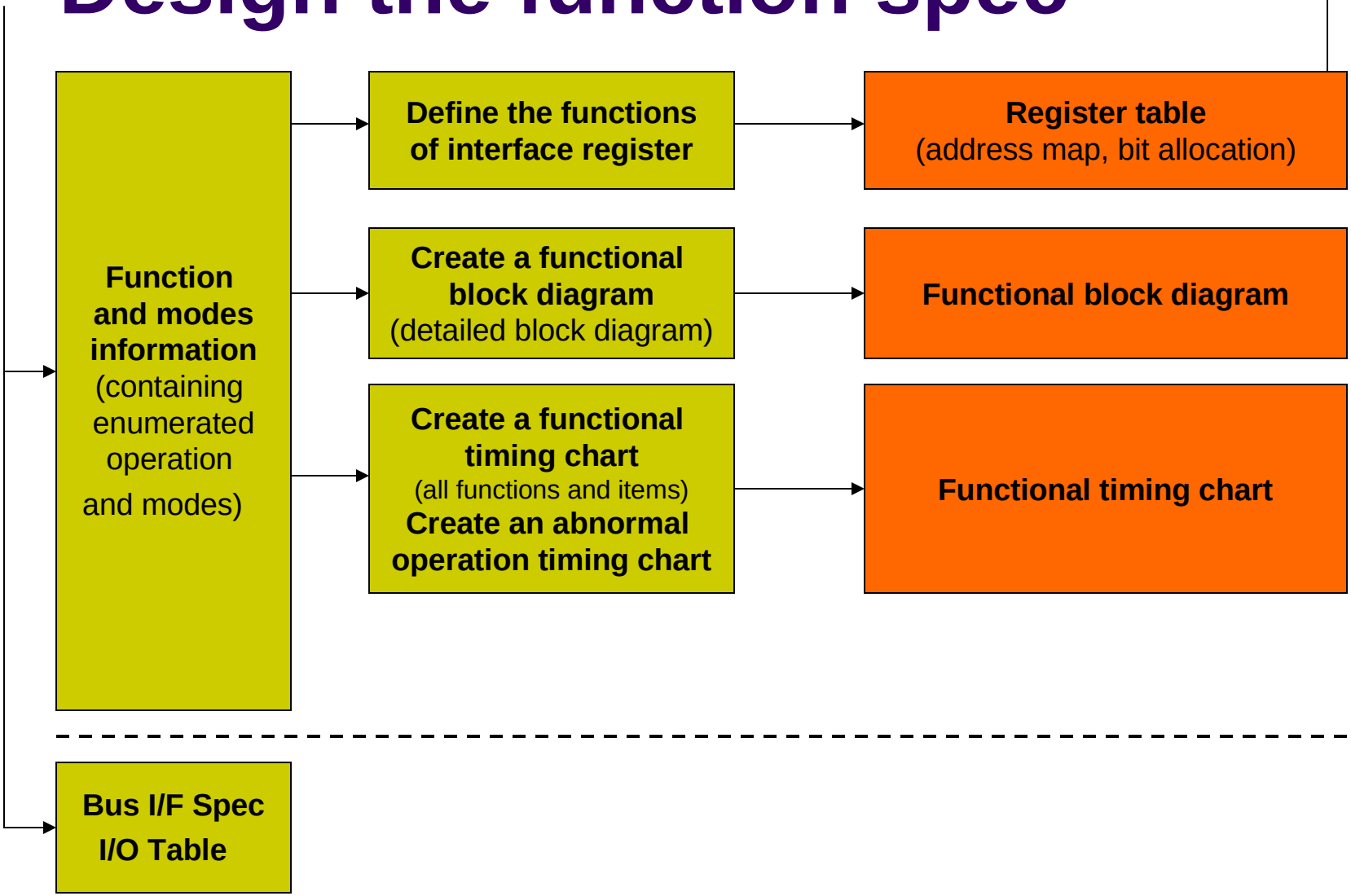
Design the function spec



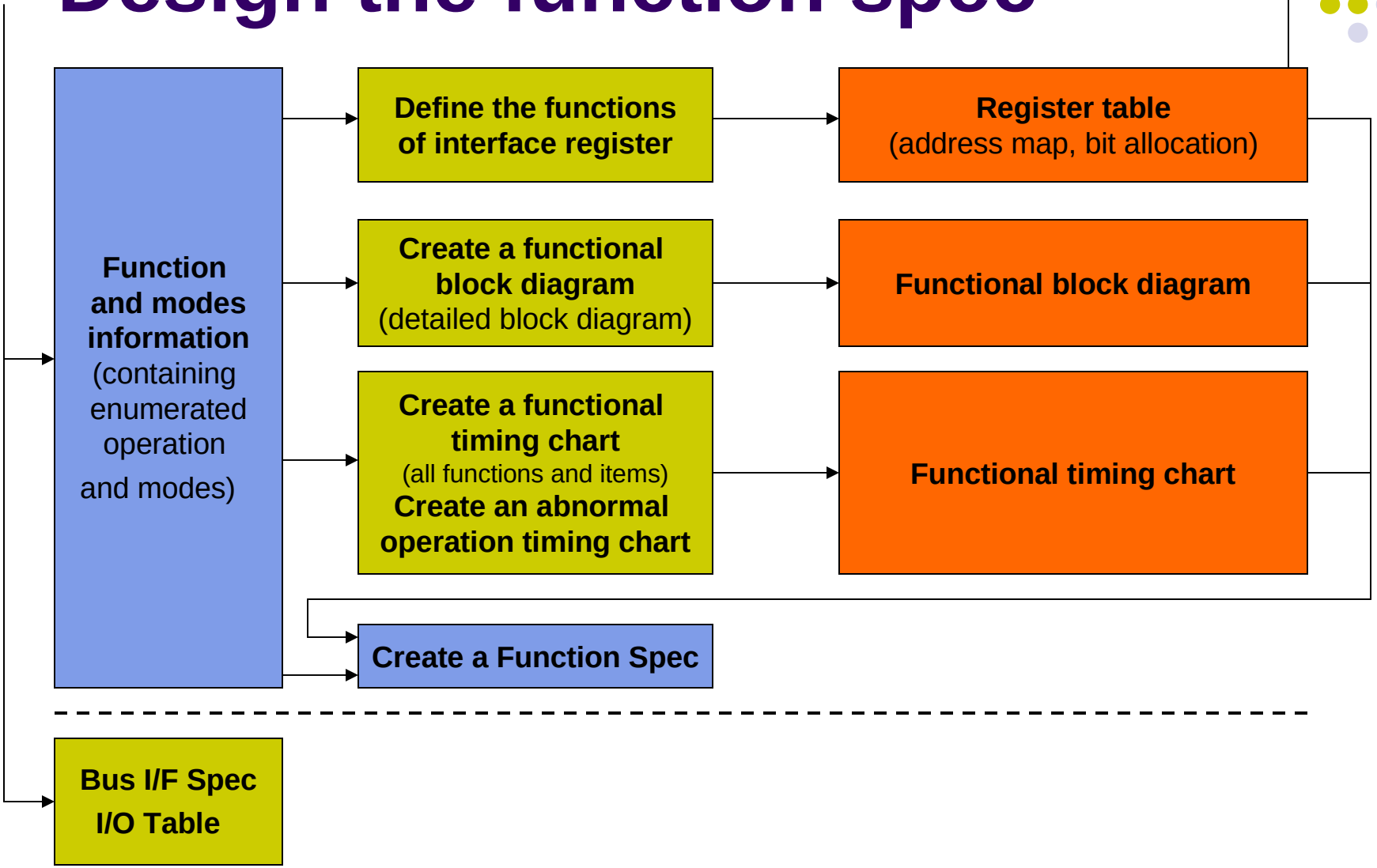
Design the function spec



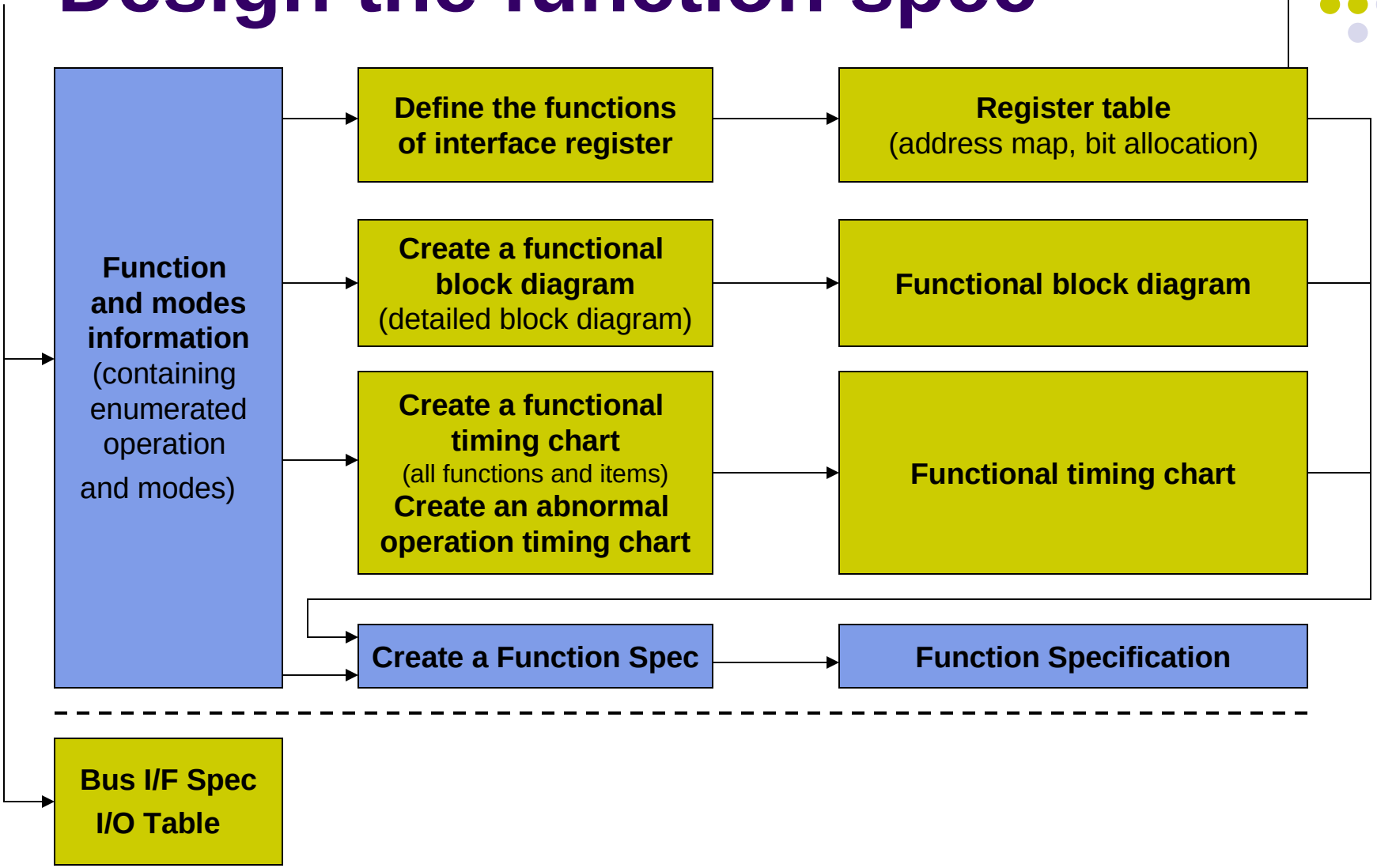
Design the function spec



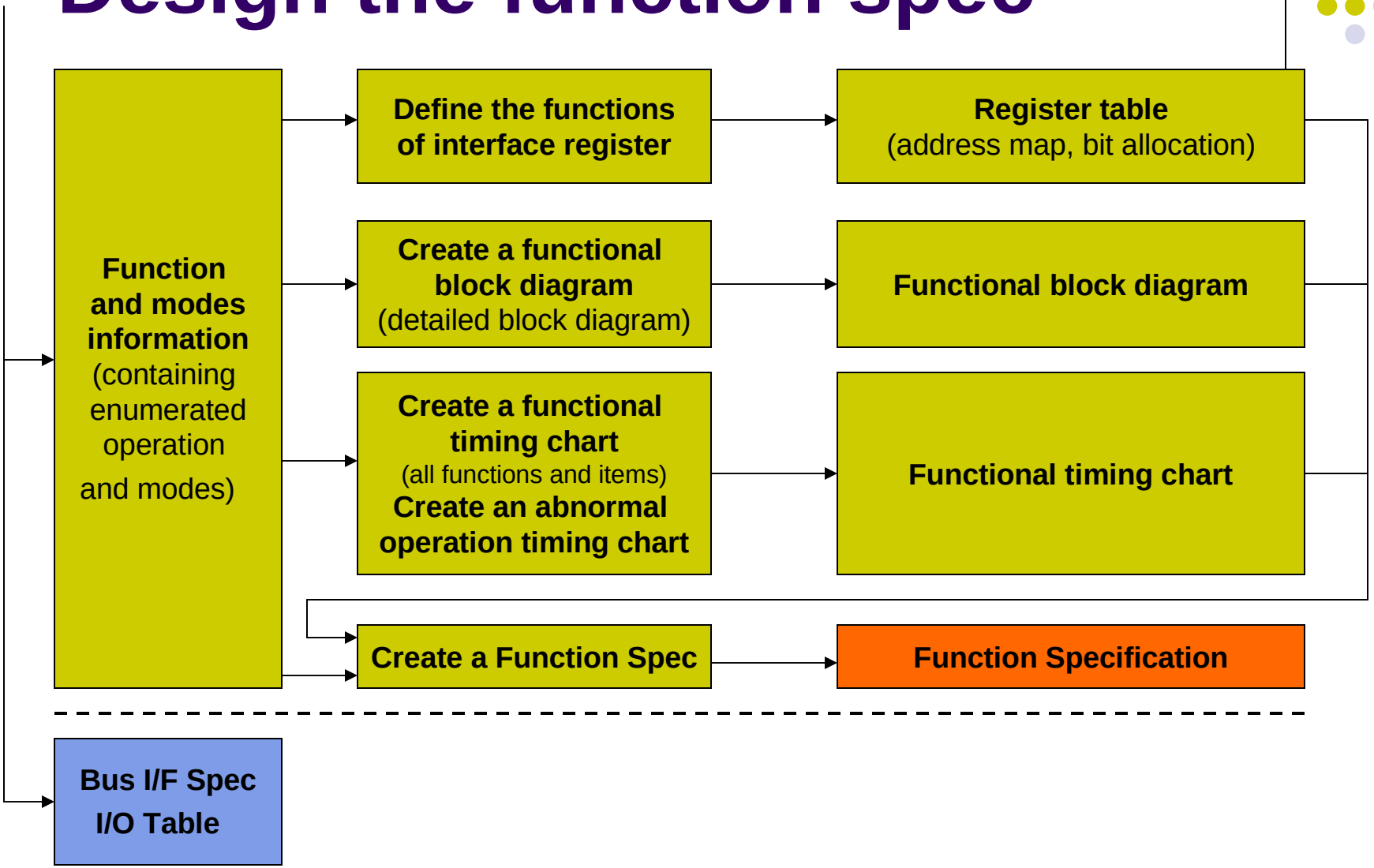
Design the function spec



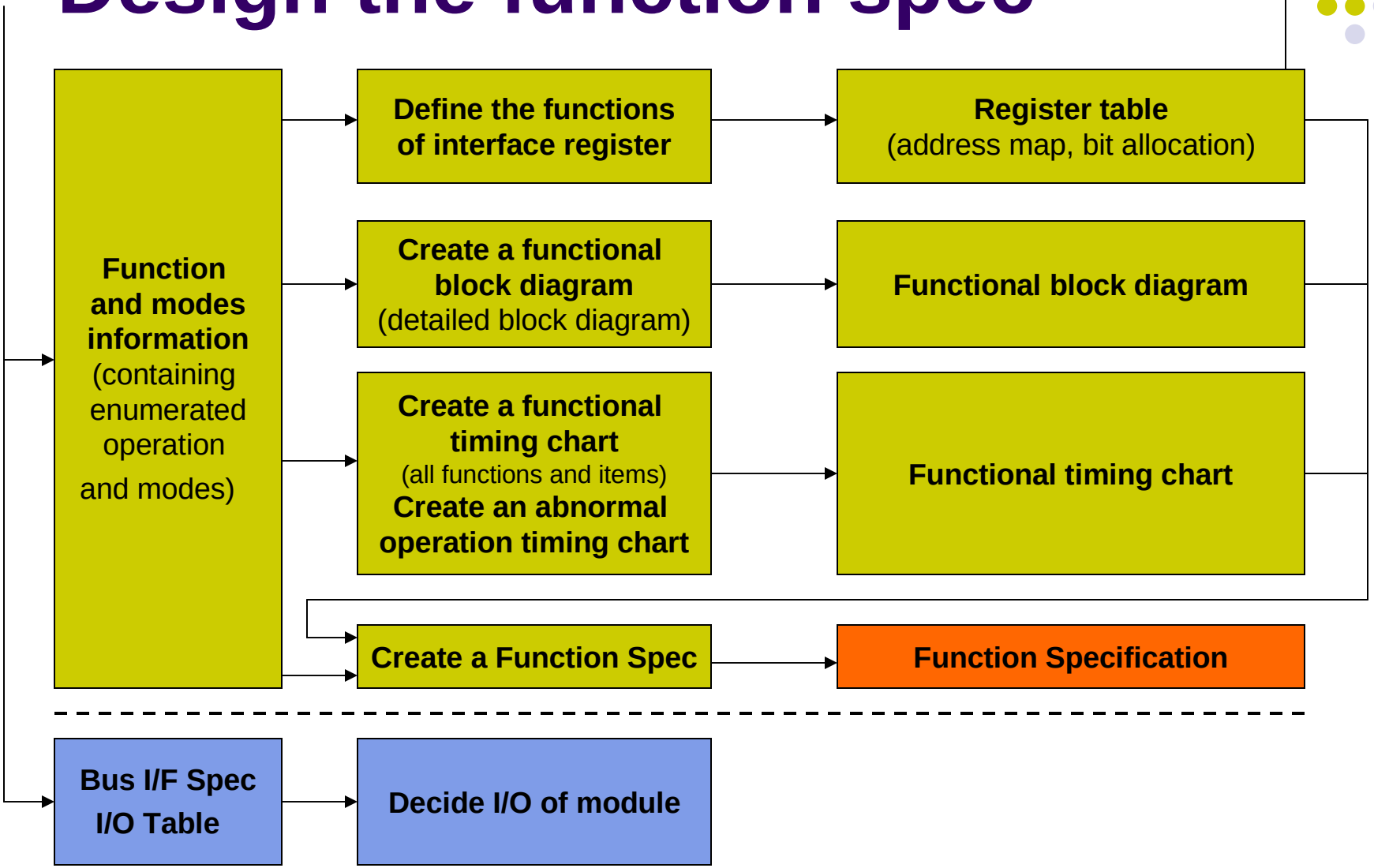
Design the function spec



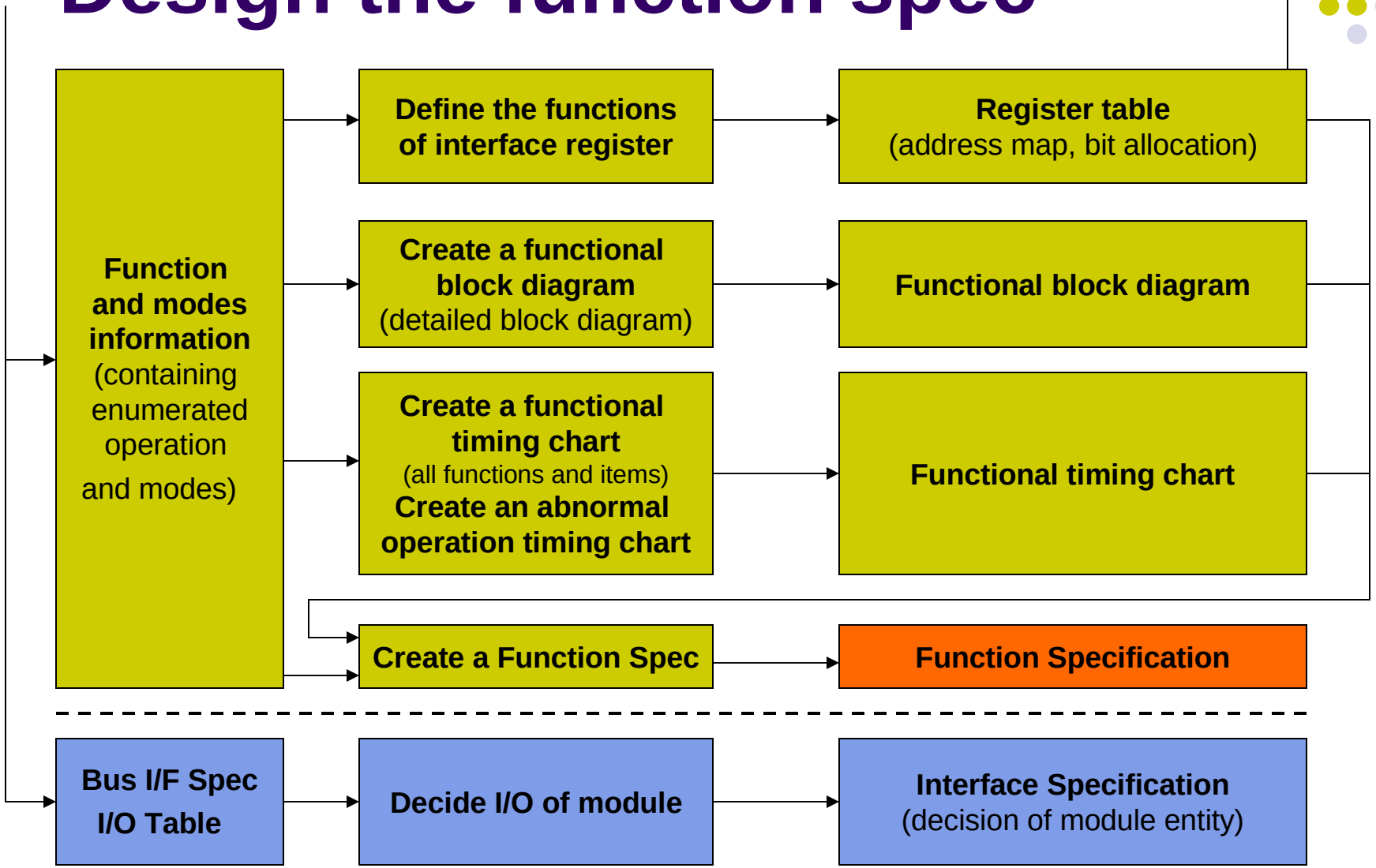
Design the function spec



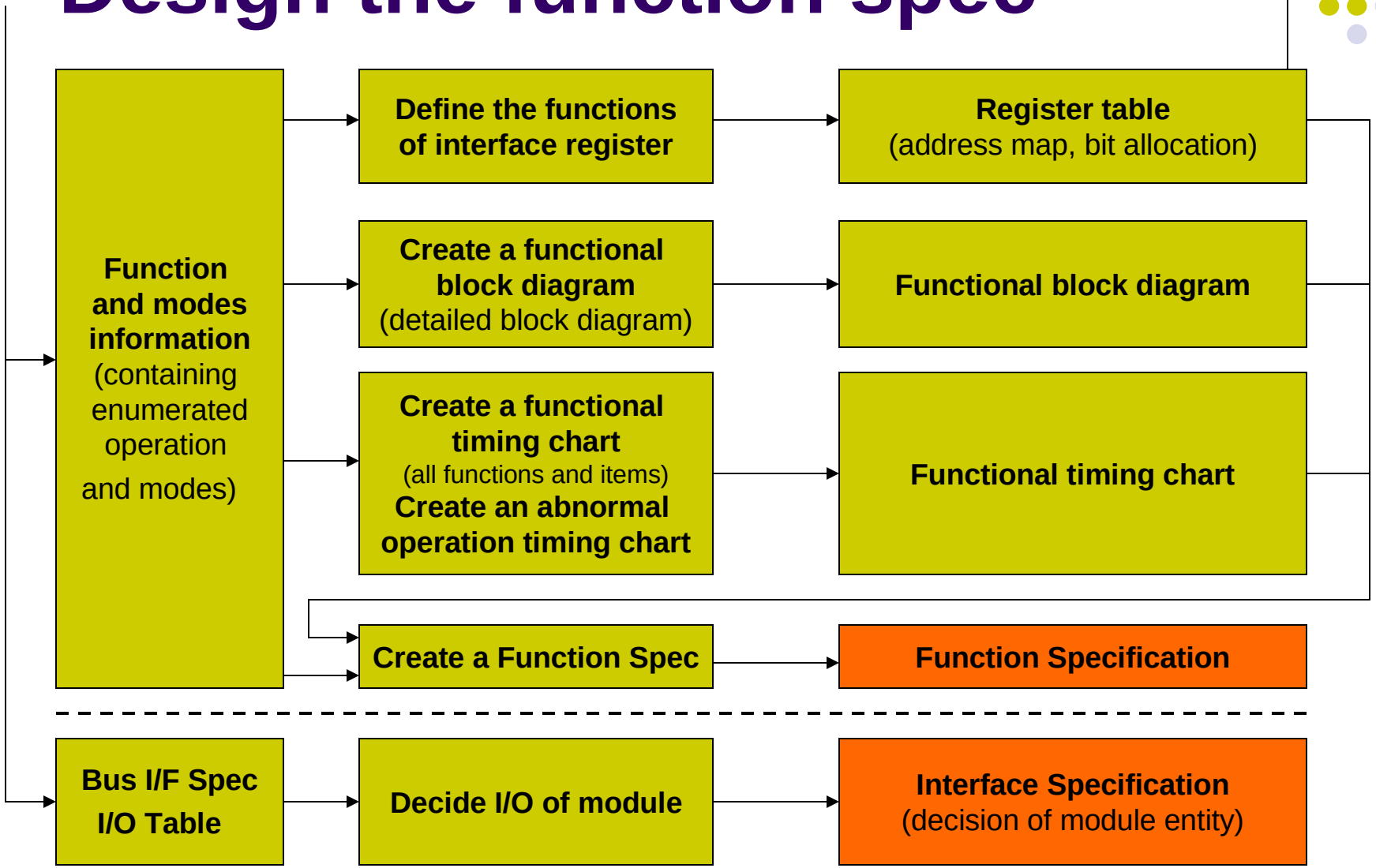
Design the function spec



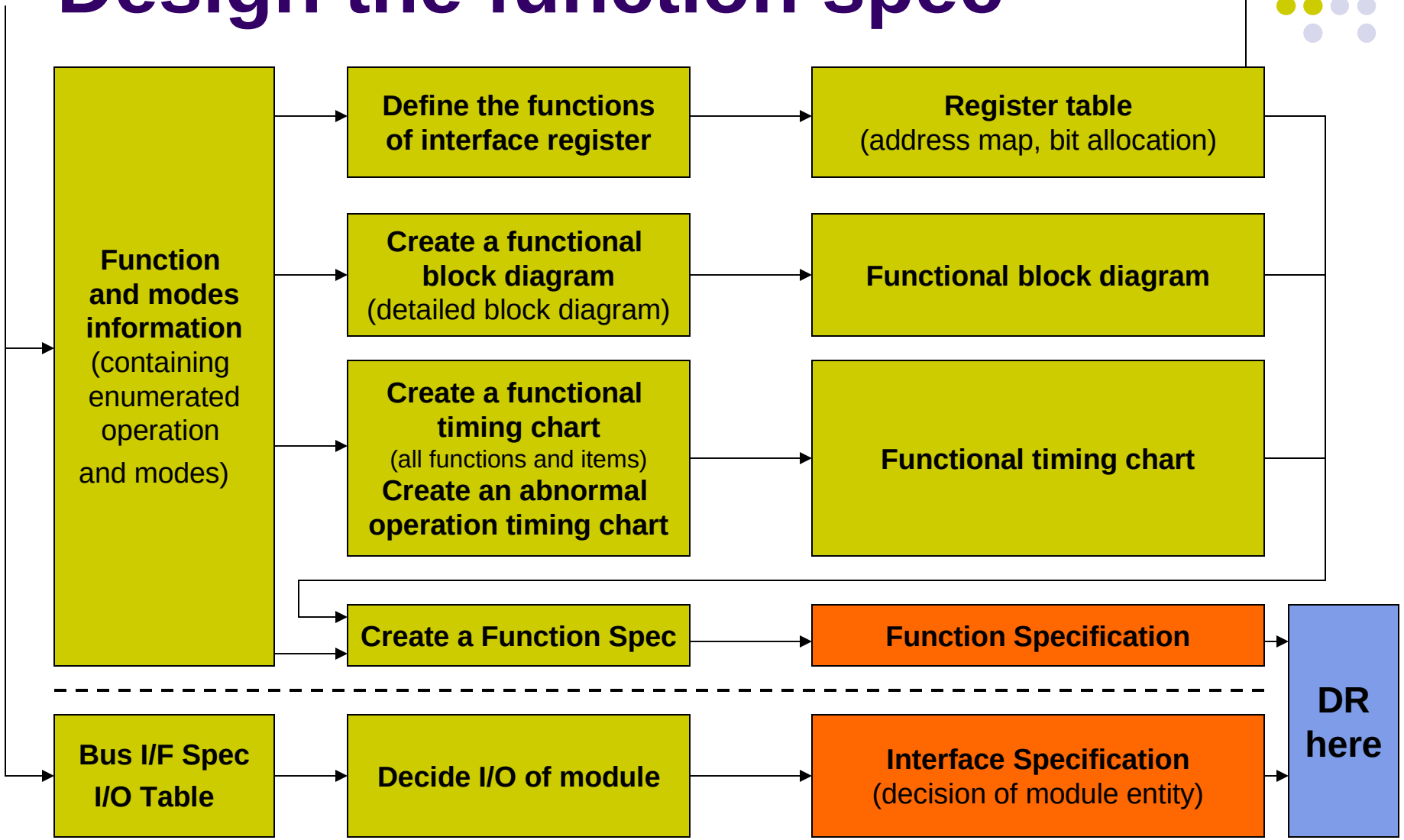
Design the function spec



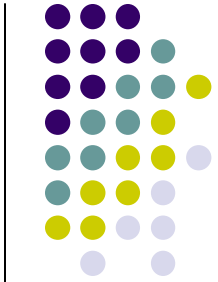
Design the function spec



Design the function spec



Create Design Details



- The module is designed detail in this process
- A detailed design is a process of deciding how to realize the function and the operation that became clearly in the process of the function specification design
- Show how to realize the function by using the block diagram and the timing chart
- Clarifying that there are neither contradiction nor considered shortage of logic
- Descryption should be easy to understand

Create Design Details

Input

- Function and modes Information (contain enumerated operation and the modes)
- Table of registers

Create Design Details

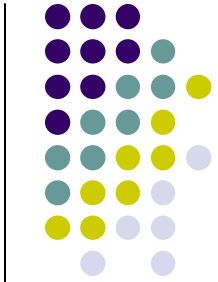
Input

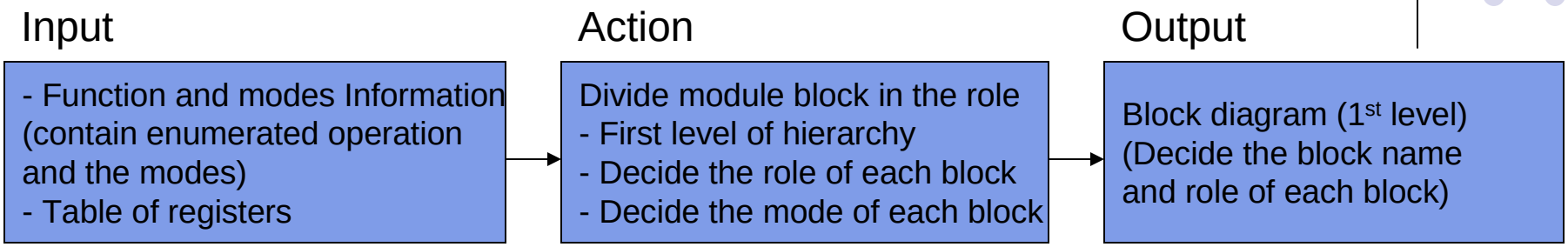
- Function and modes Information (contain enumerated operation and the modes)
- Table of registers

Action

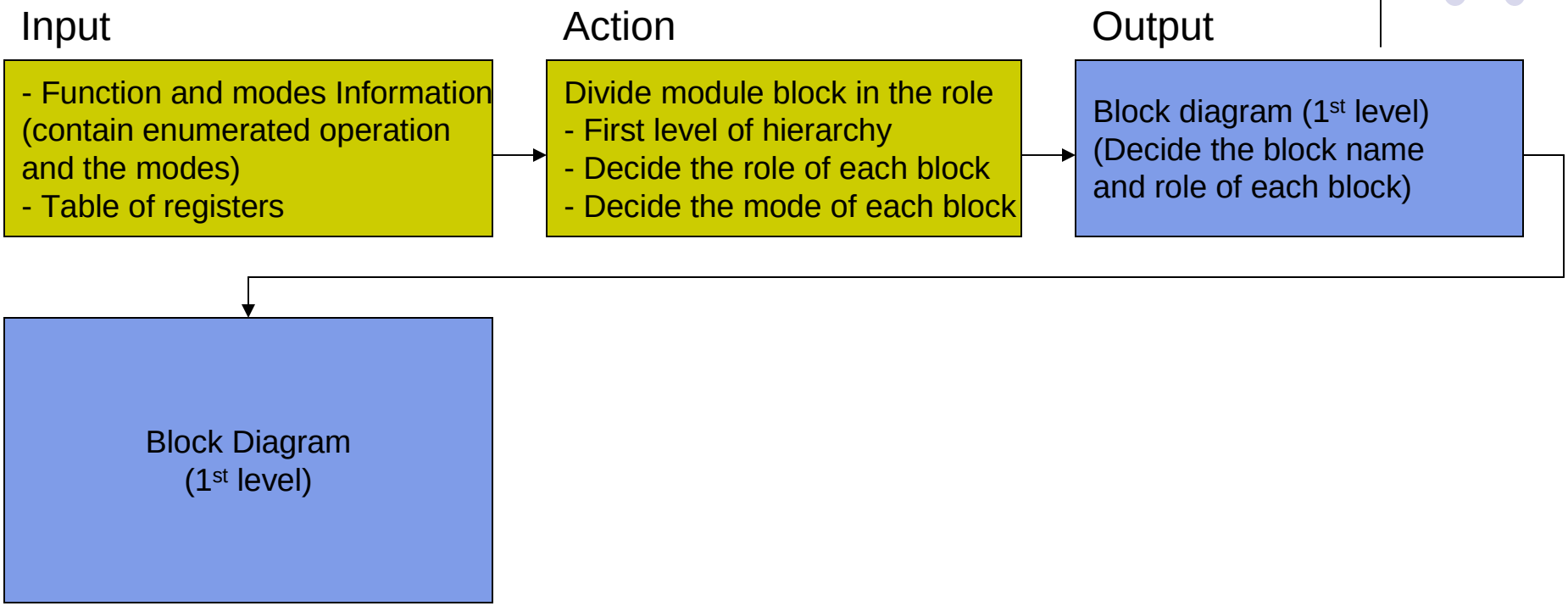
Divide module block in the role
- First level of hierarchy
- Decide the role of each block
- Decide the mode of each block

Create Design Details

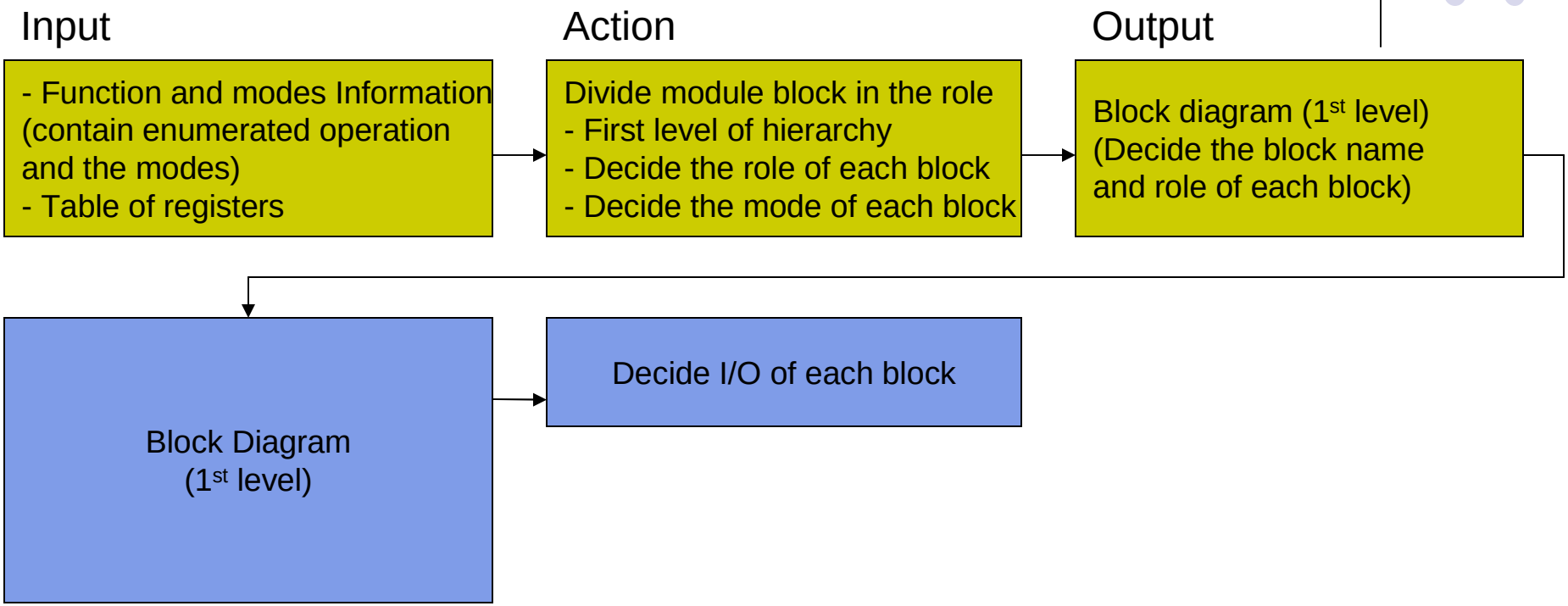




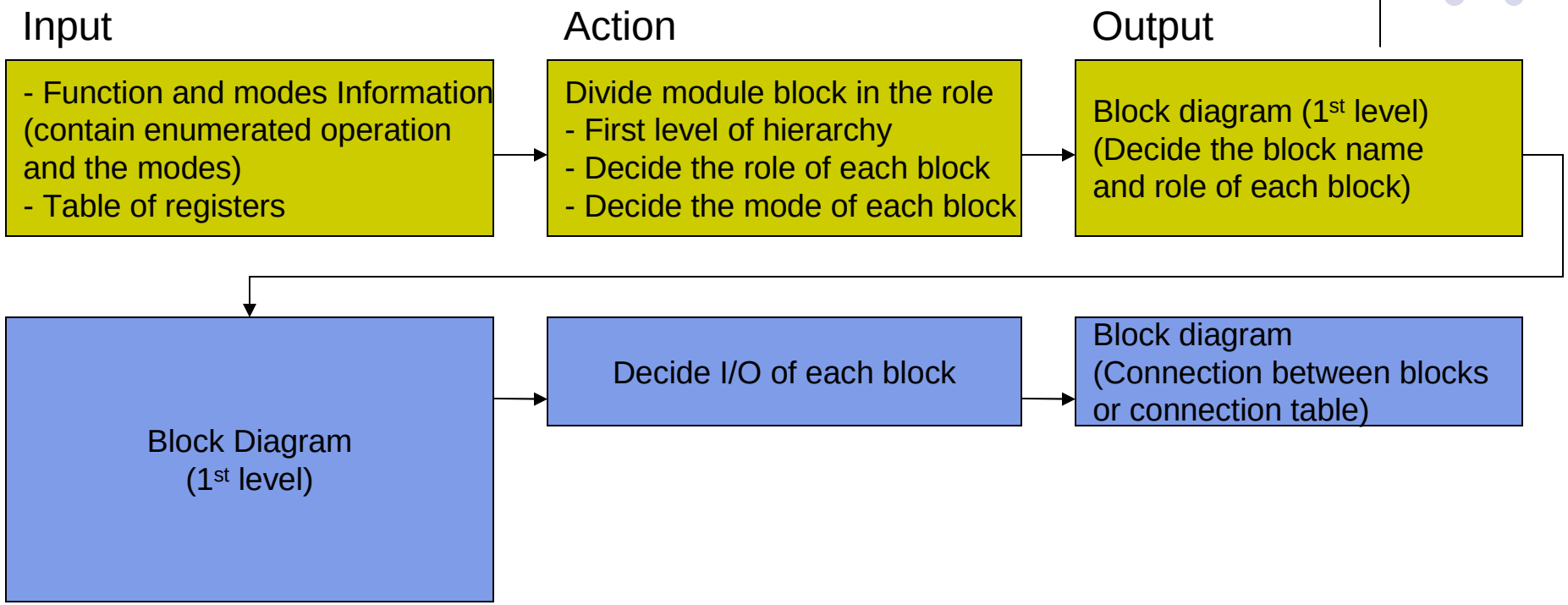
Create Design Details



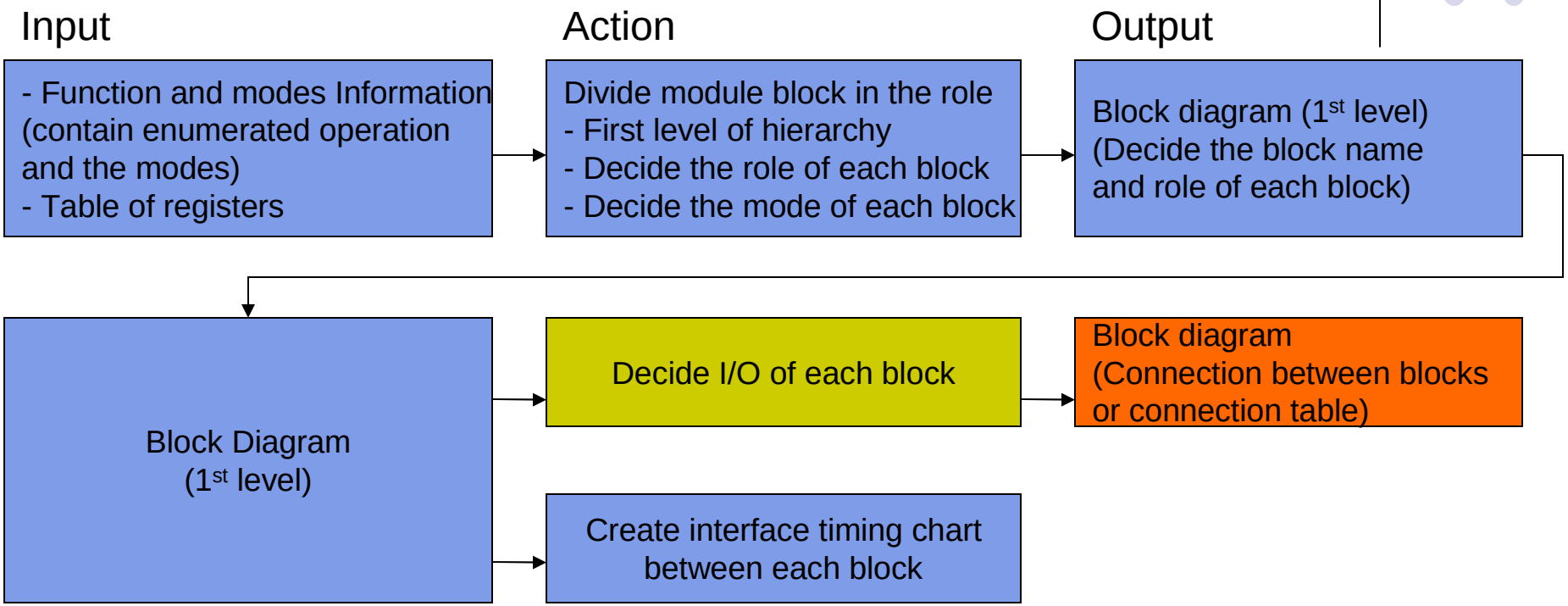
Create Design Details



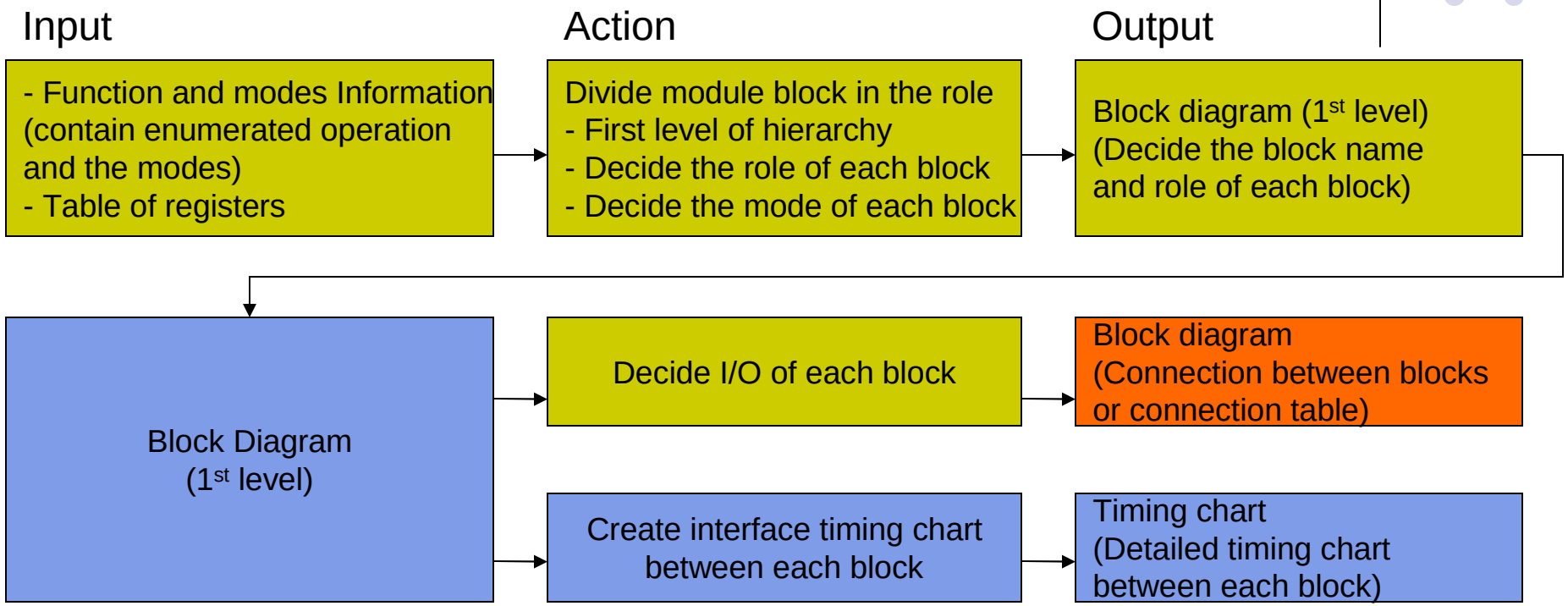
Create Design Details



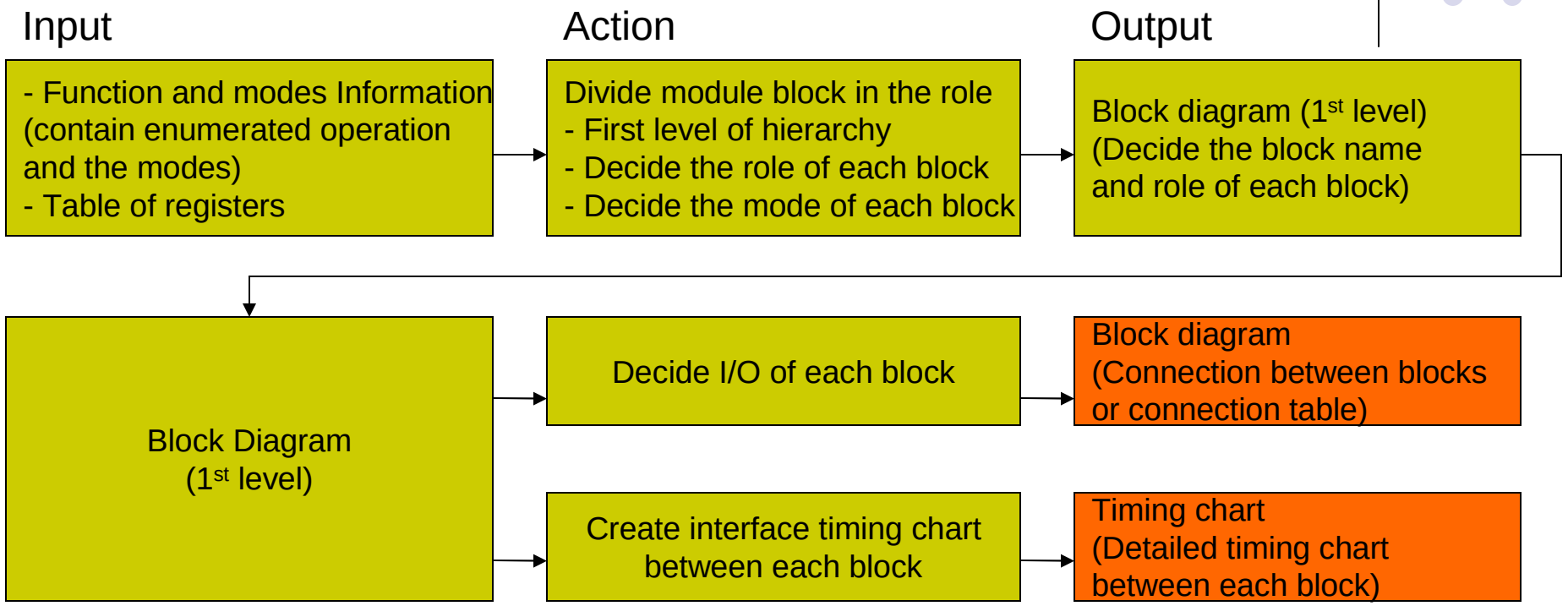
Create Design Details



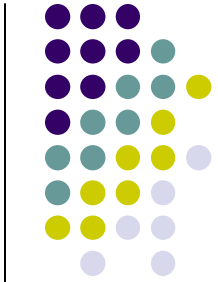
Create Design Details

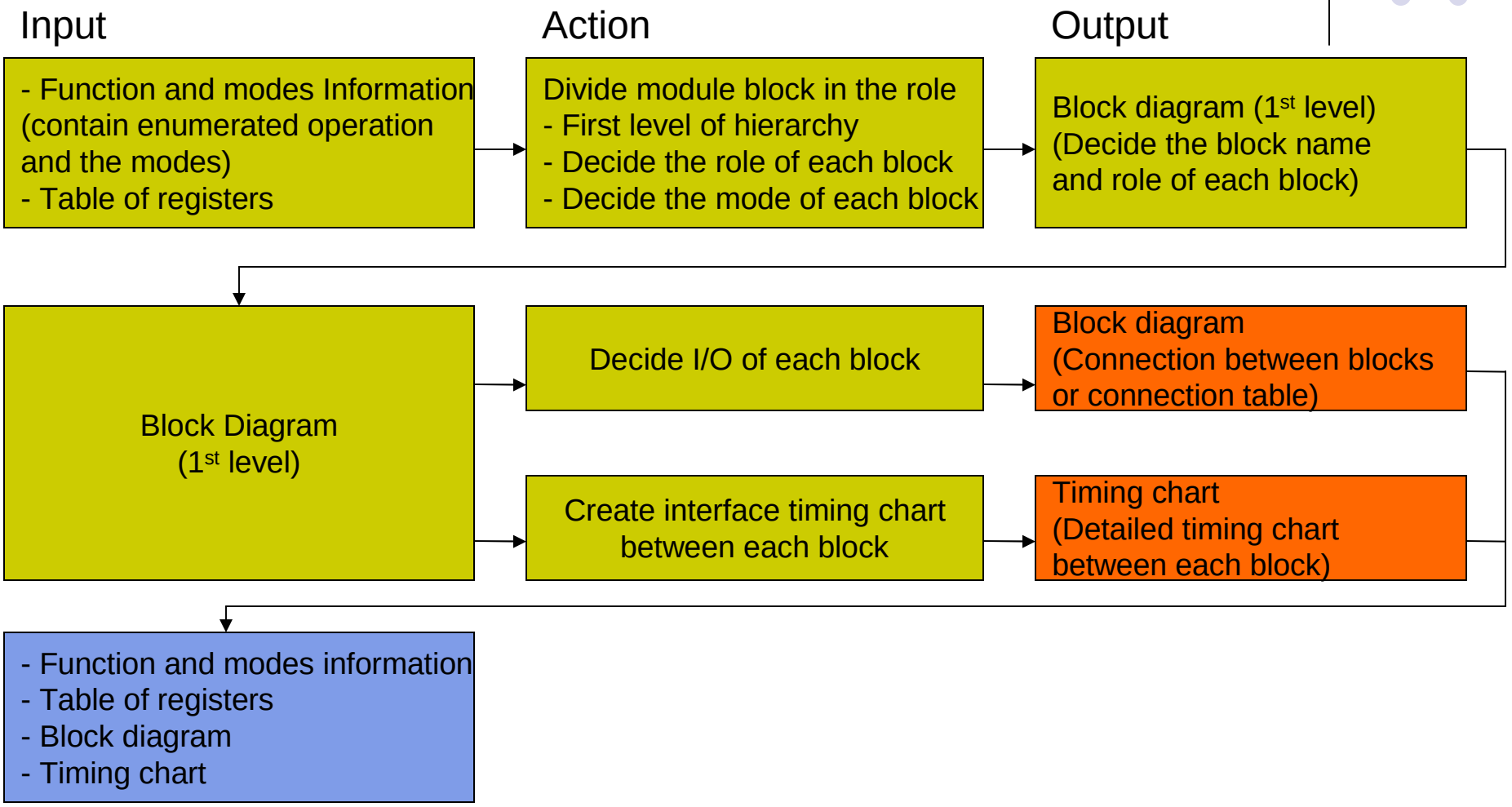


Create Design Details

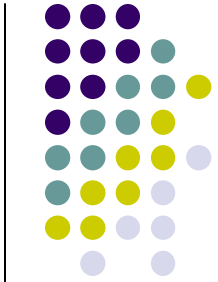


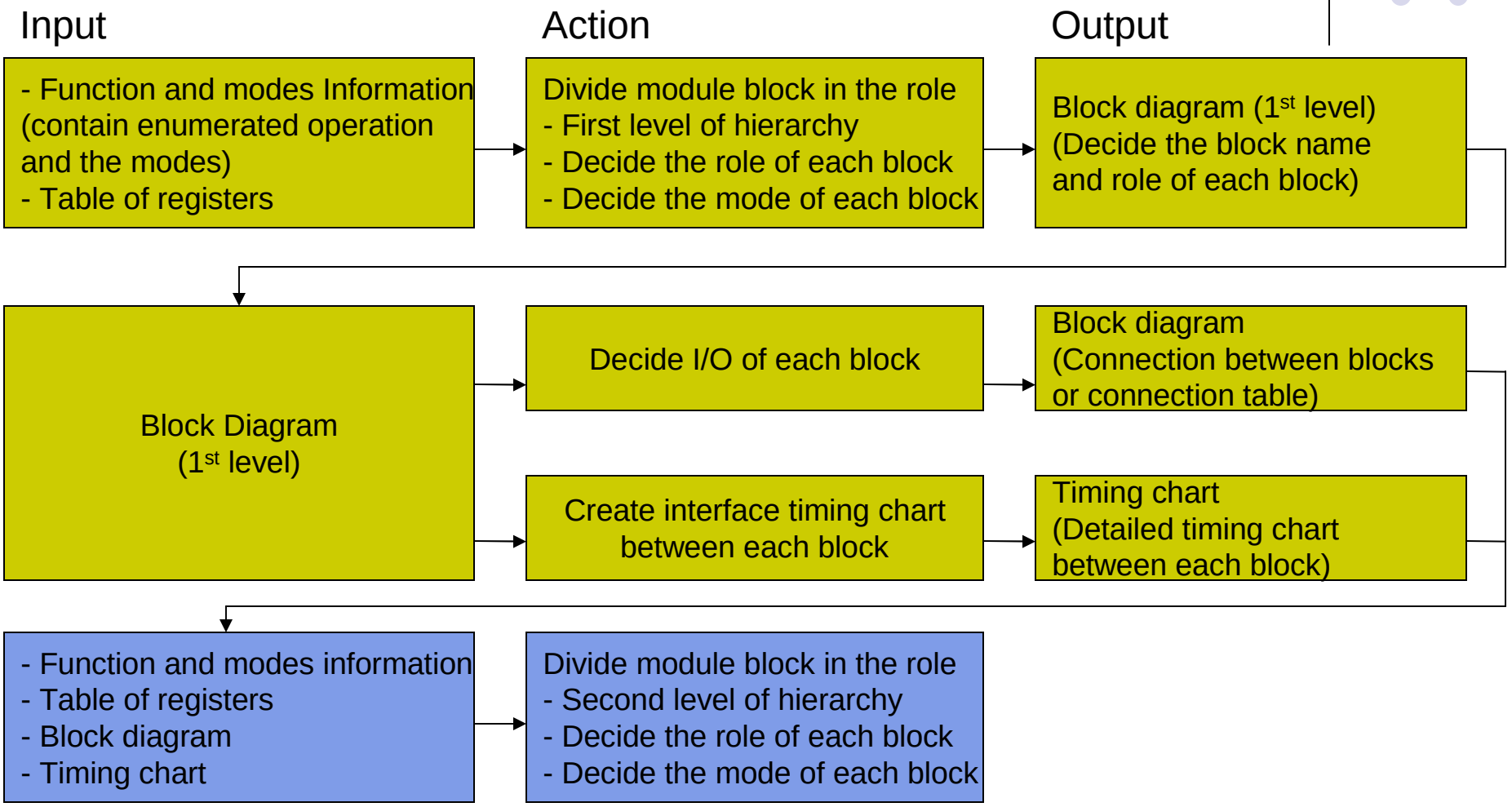
Create Design Details



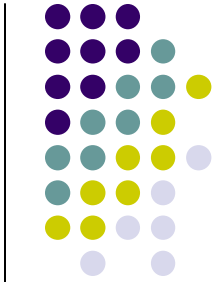


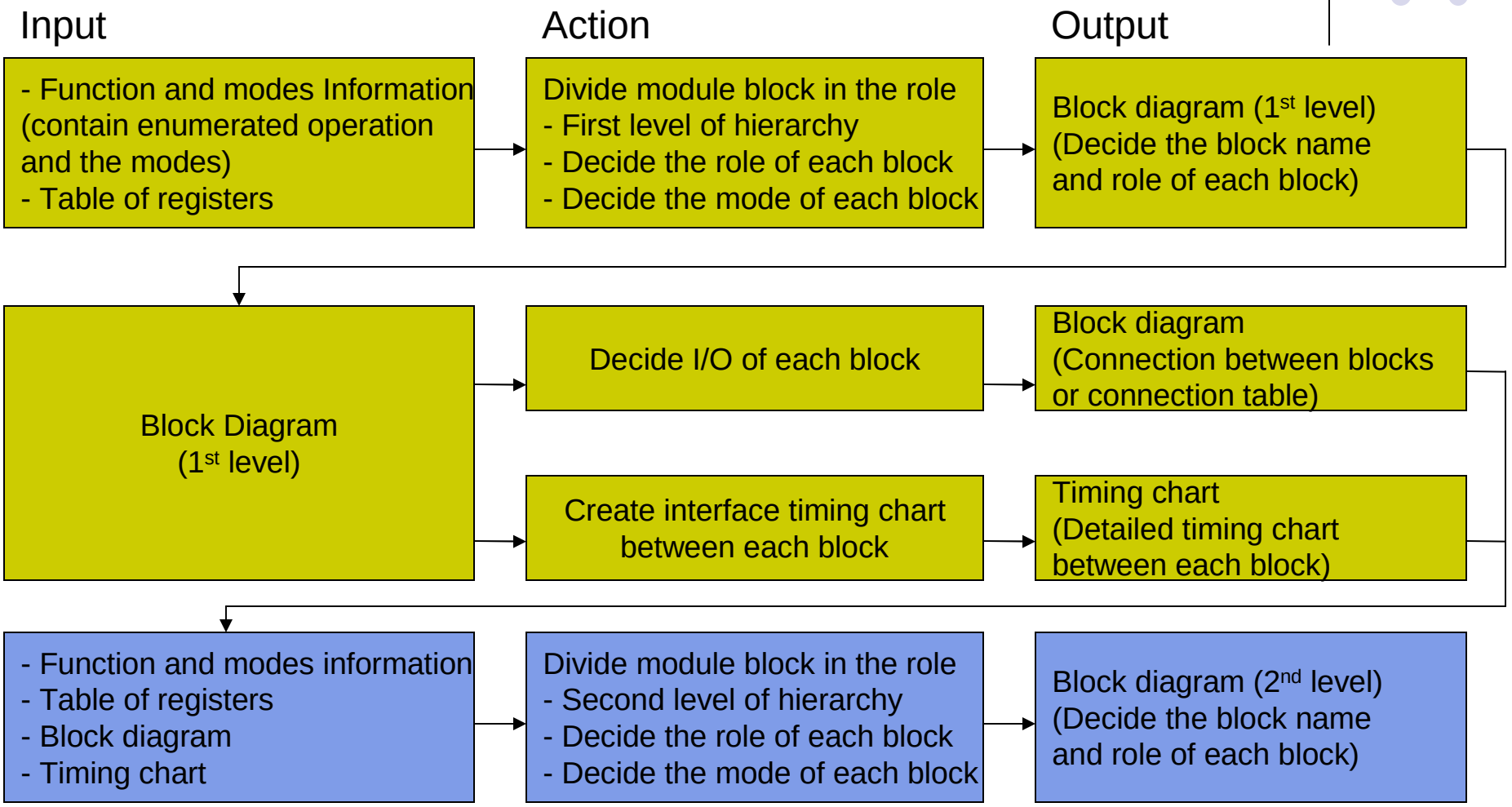
Create Design Details



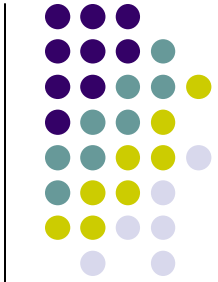


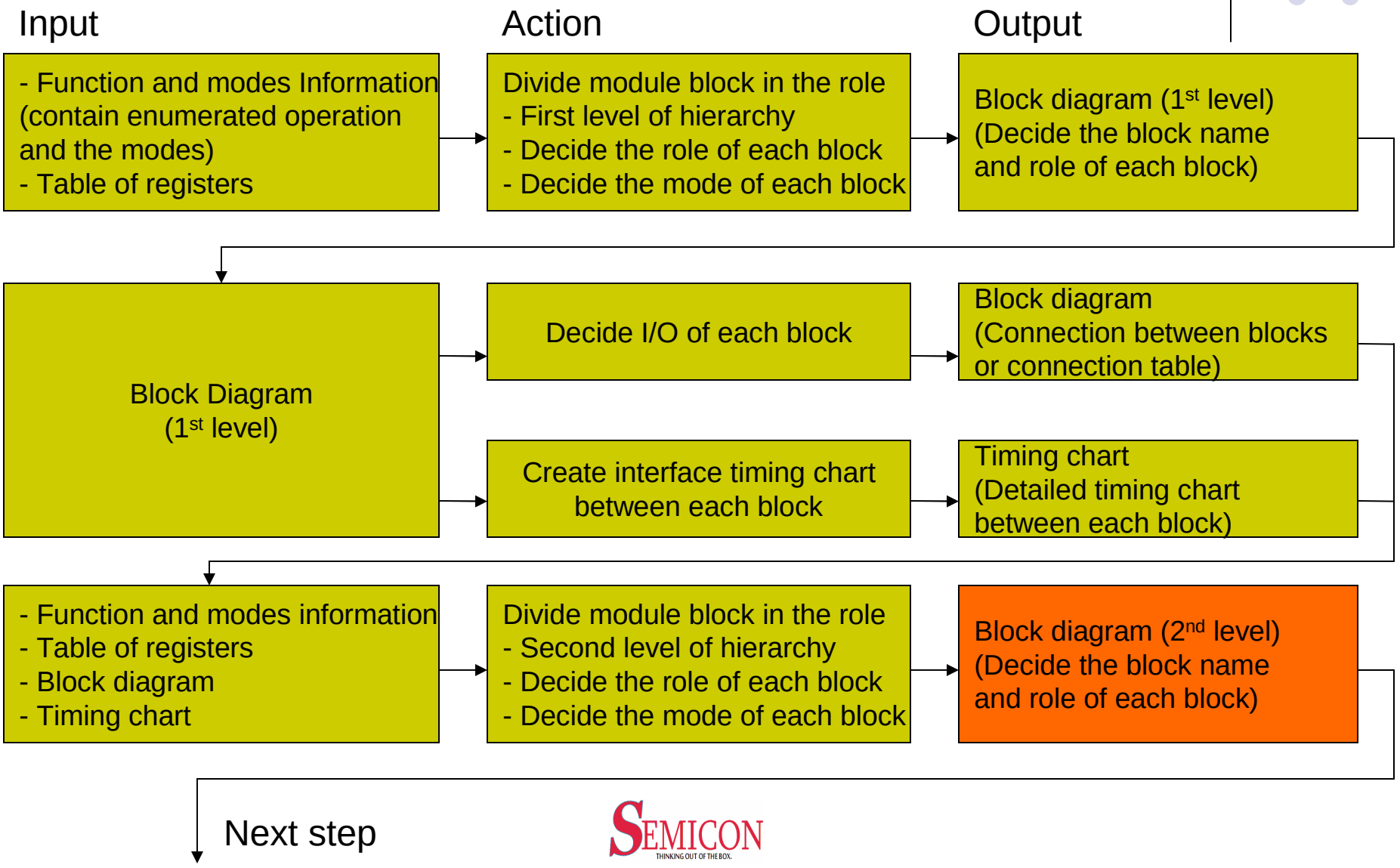
Create Design Details





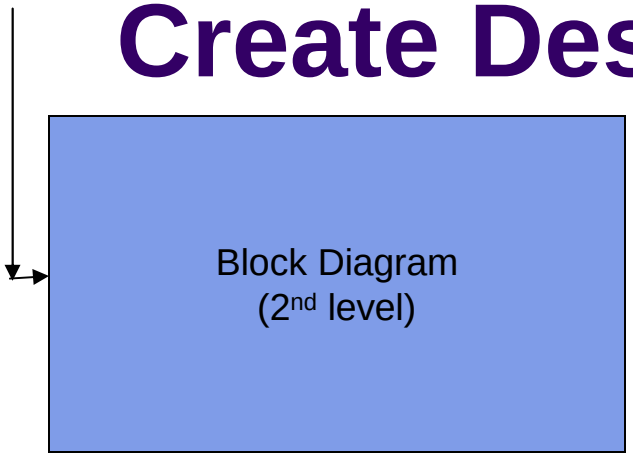
Create Design Details



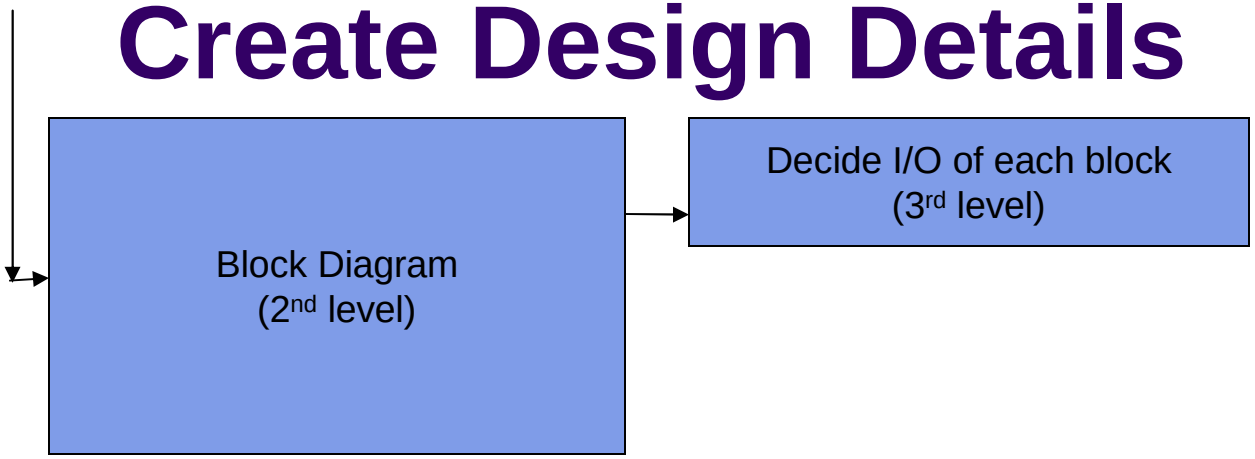


SEMICON
THINKING OUT OF THE BOX.

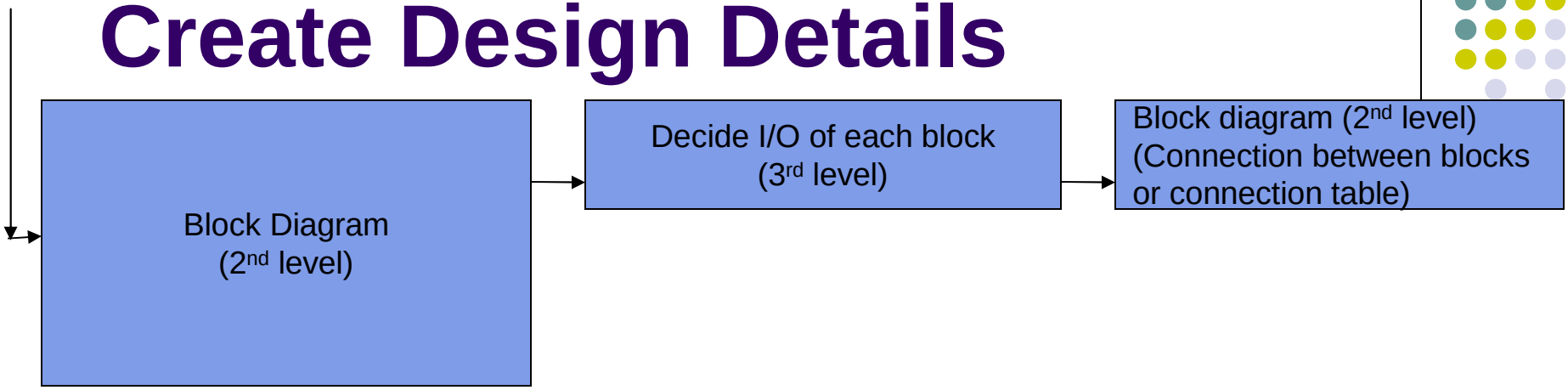
Create Design Details



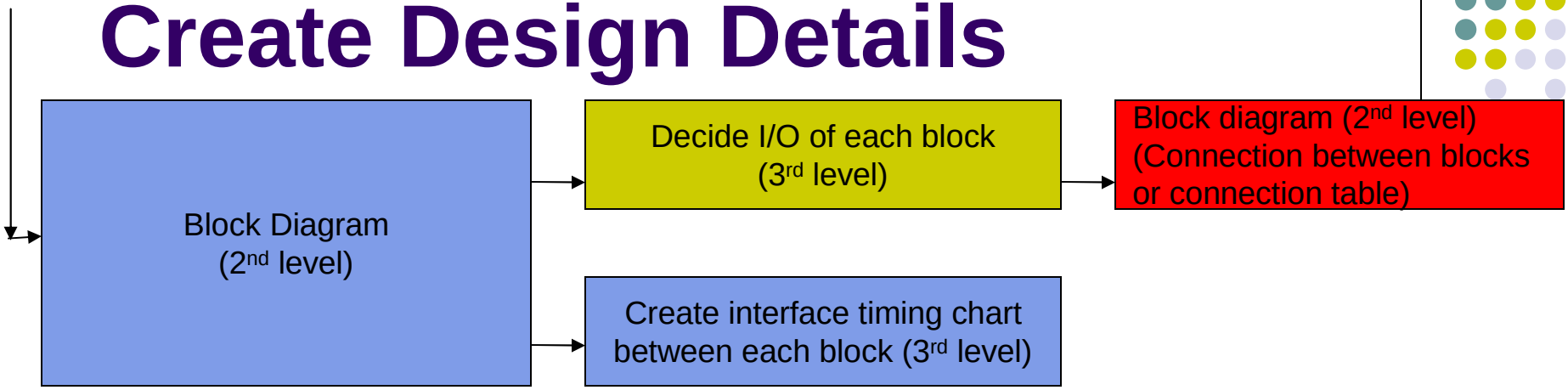
Create Design Details



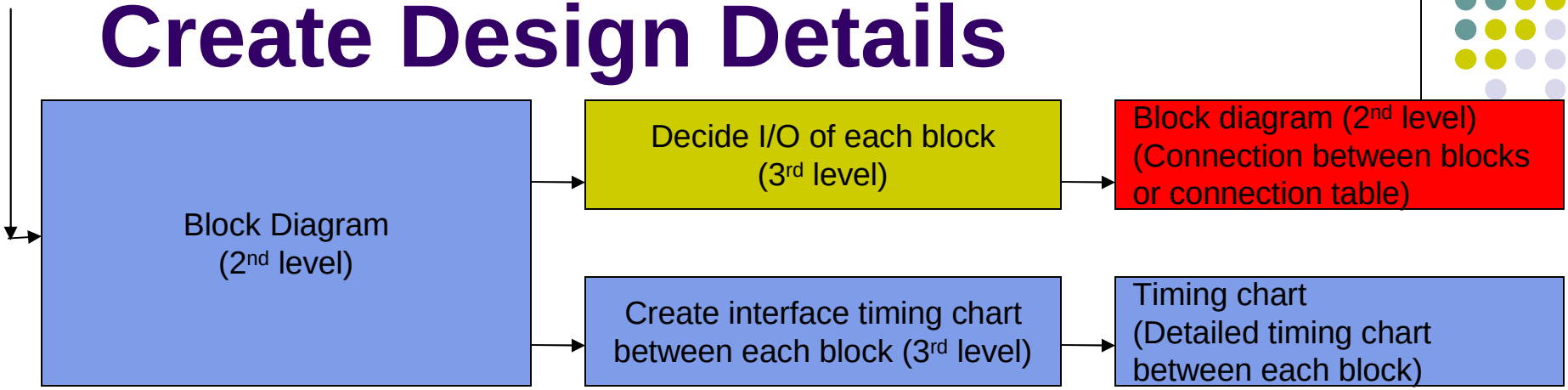
Create Design Details



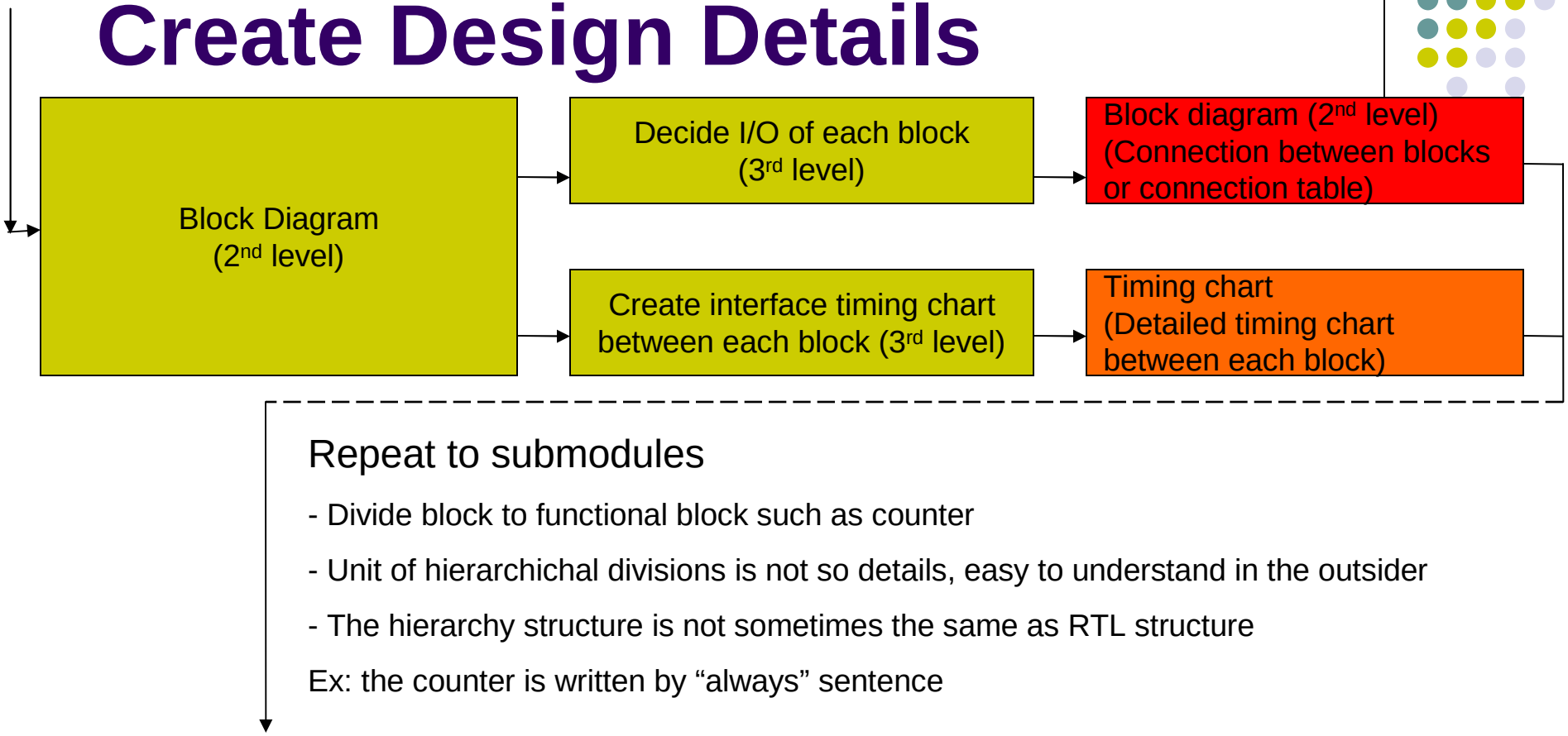
Create Design Details



Create Design Details



Create Design Details

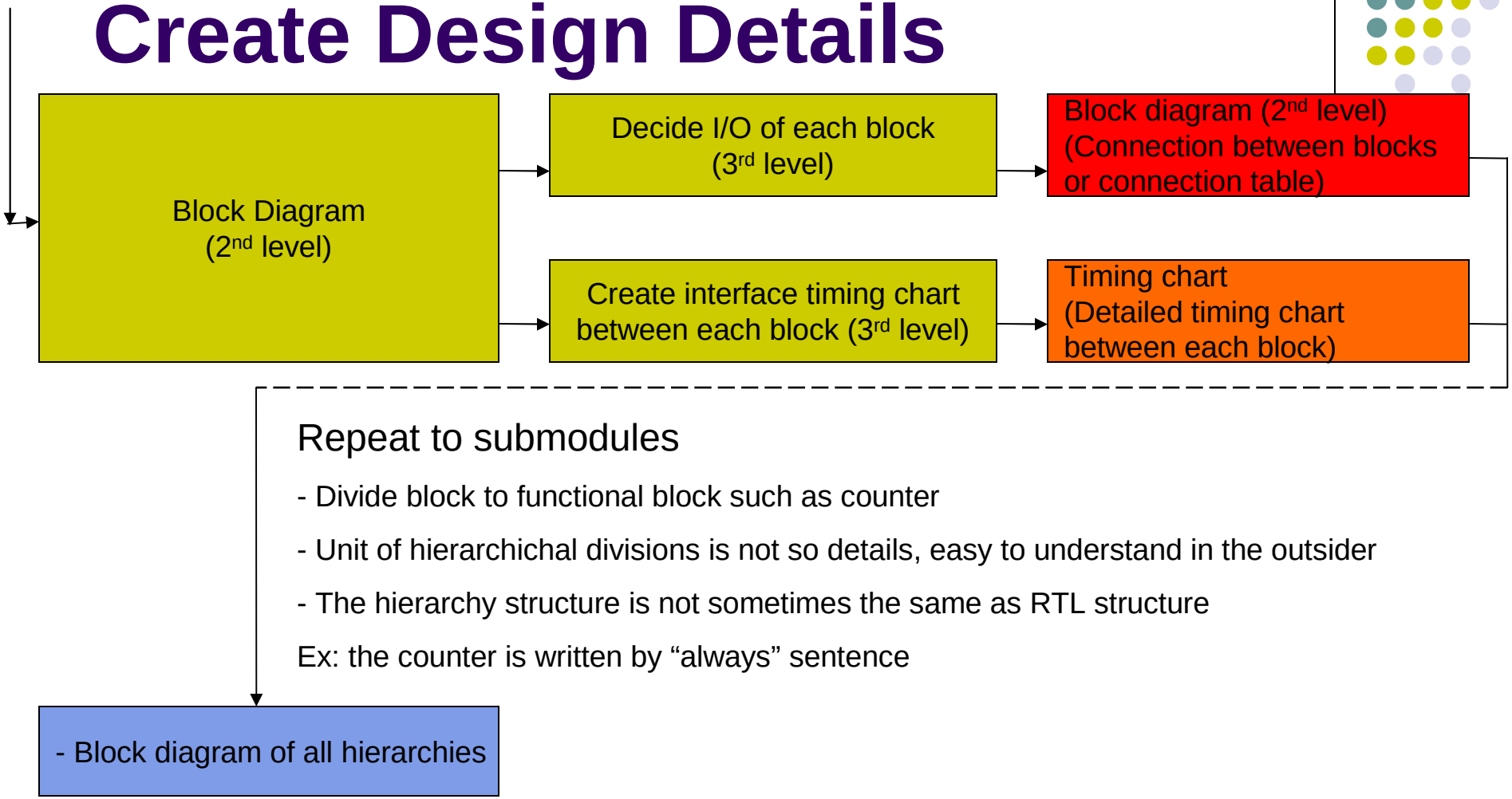


Repeat to submodules

- Divide block to functional block such as counter
- Unit of hierarchichal divisions is not so details, easy to understand in the outsider
- The hierarchy structure is not sometimes the same as RTL structure

Ex: the counter is written by “always” sentence

Create Design Details



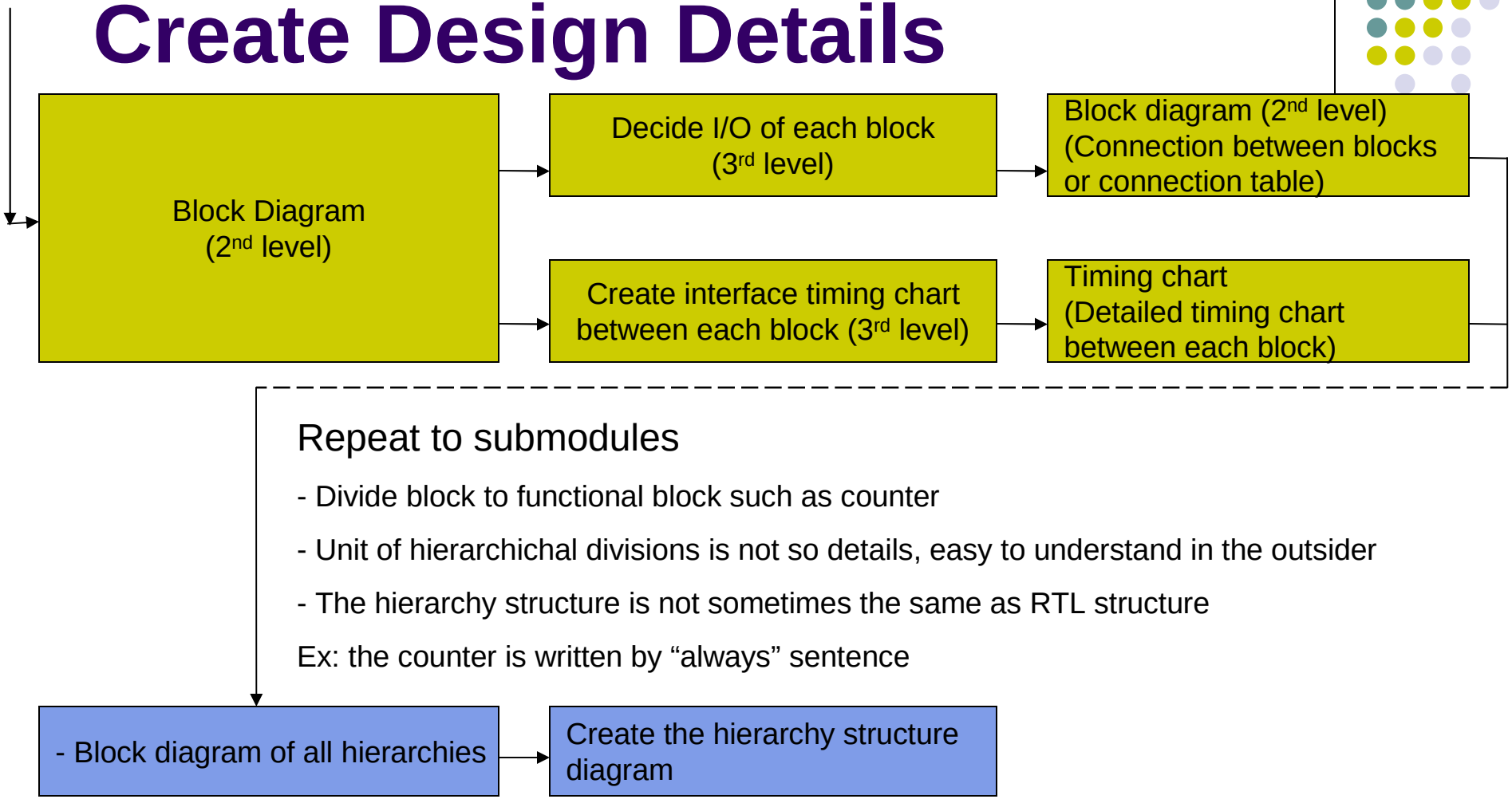
Repeat to submodules

- Divide block to functional block such as counter
- Unit of hierarchichal divisions is not so details, easy to understand in the outsider
- The hierarchy structure is not sometimes the same as RTL structure

Ex: the counter is written by “always” sentence

- Block diagram of all hierarchies

Create Design Details

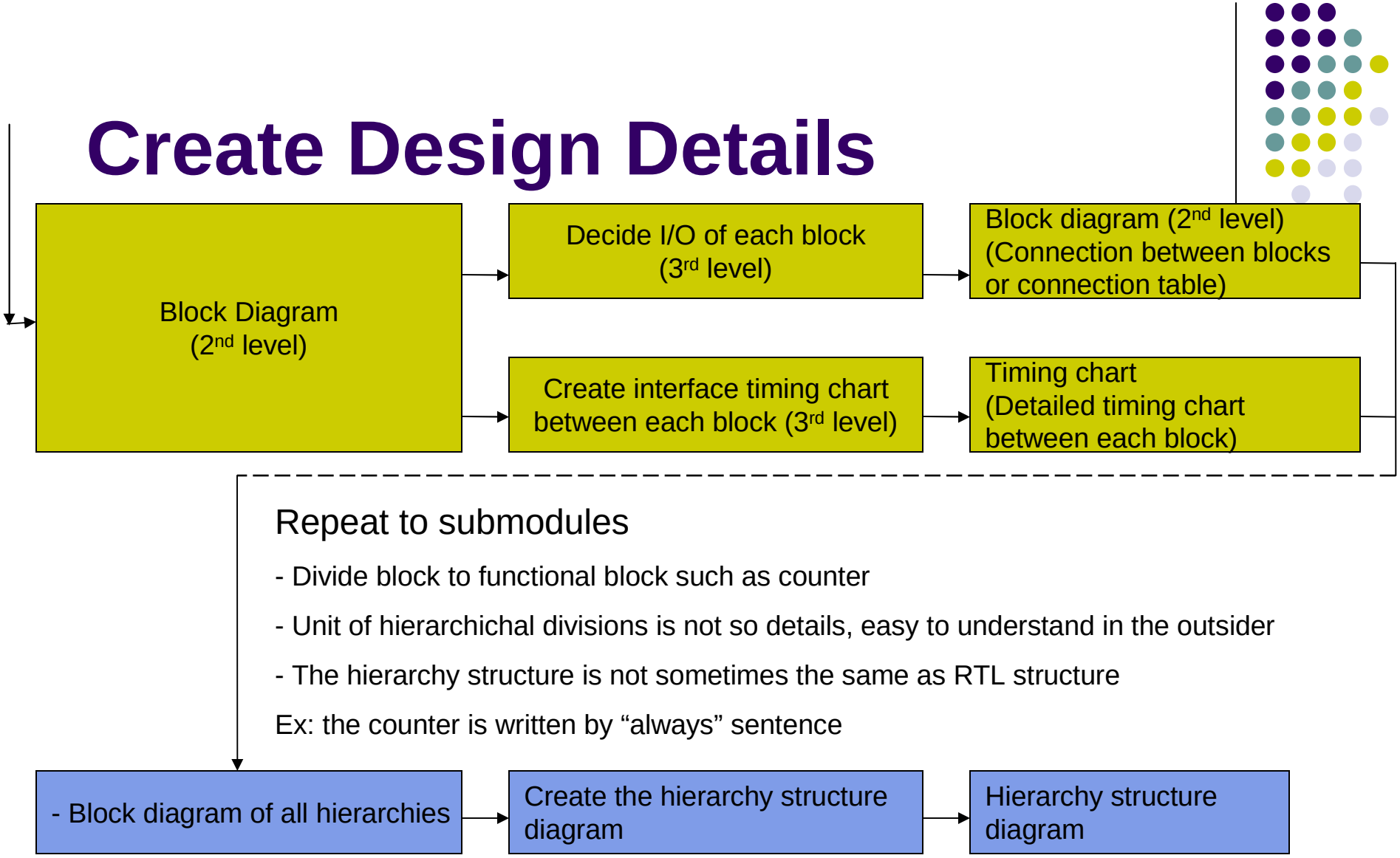


Repeat to submodules

- Divide block to functional block such as counter
- Unit of hierarchichal divisions is not so details, easy to understand in the outsider
- The hierarchy structure is not sometimes the same as RTL structure

Ex: the counter is written by “always” sentence

Create Design Details

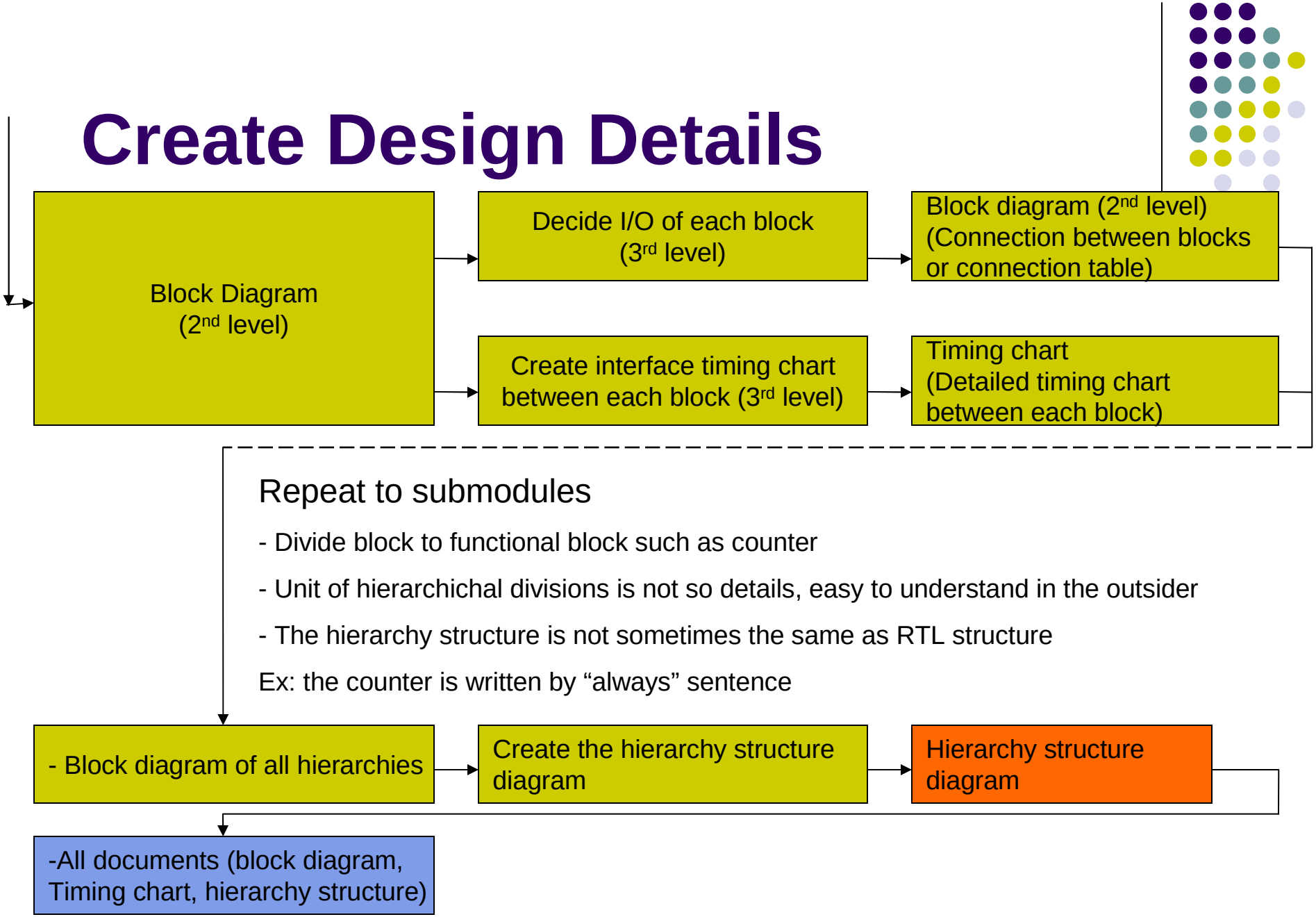


Repeat to submodules

- Divide block to functional block such as counter
- Unit of hierarchichal divisions is not so details, easy to understand in the outsider
- The hierarchy structure is not sometimes the same as RTL structure

Ex: the counter is written by “always” sentence

Create Design Details

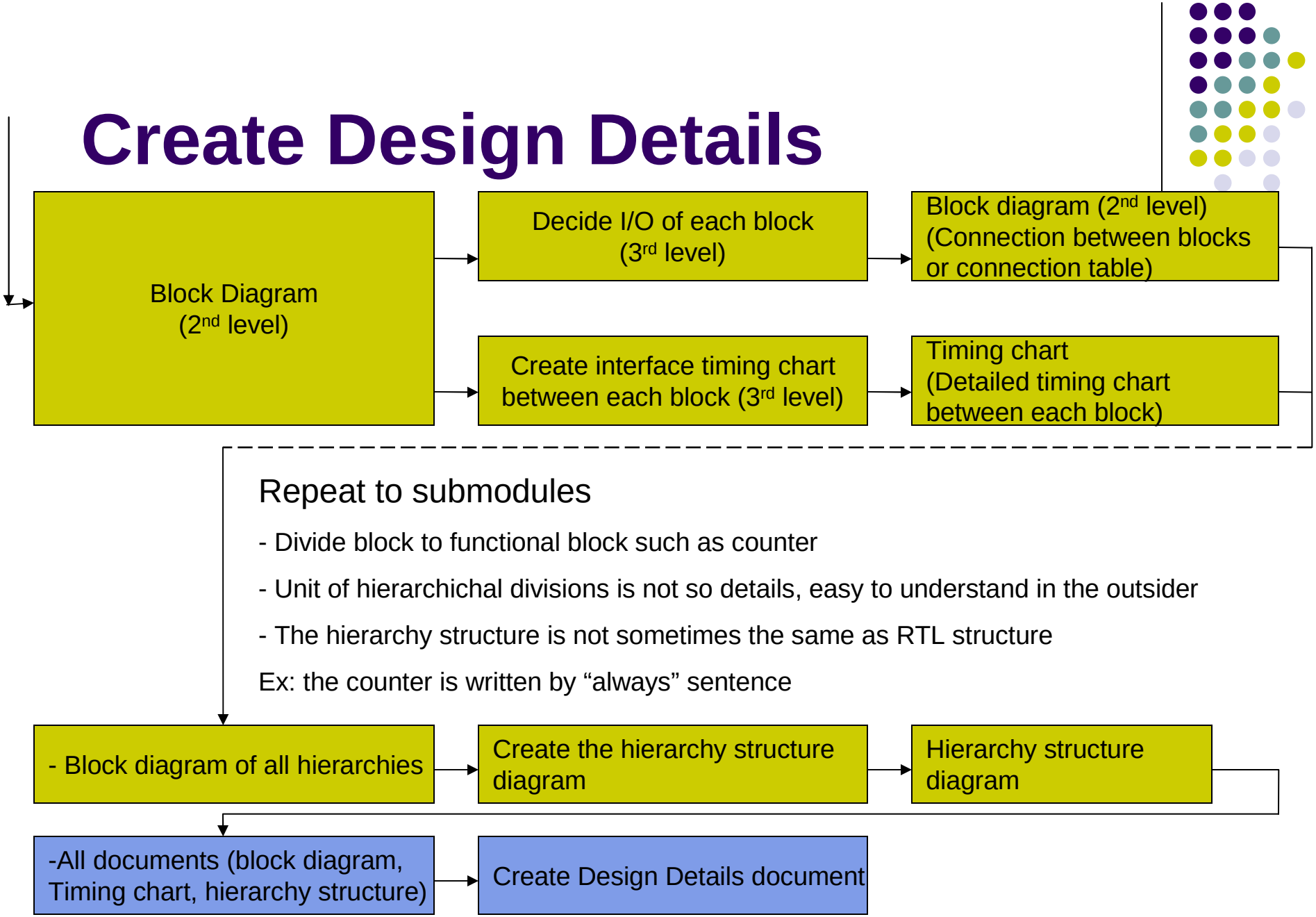


Repeat to submodules

- Divide block to functional block such as counter
- Unit of hierarchichal divisions is not so details, easy to understand in the outsider
- The hierarchy structure is not sometimes the same as RTL structure

Ex: the counter is written by “always” sentence

Create Design Details



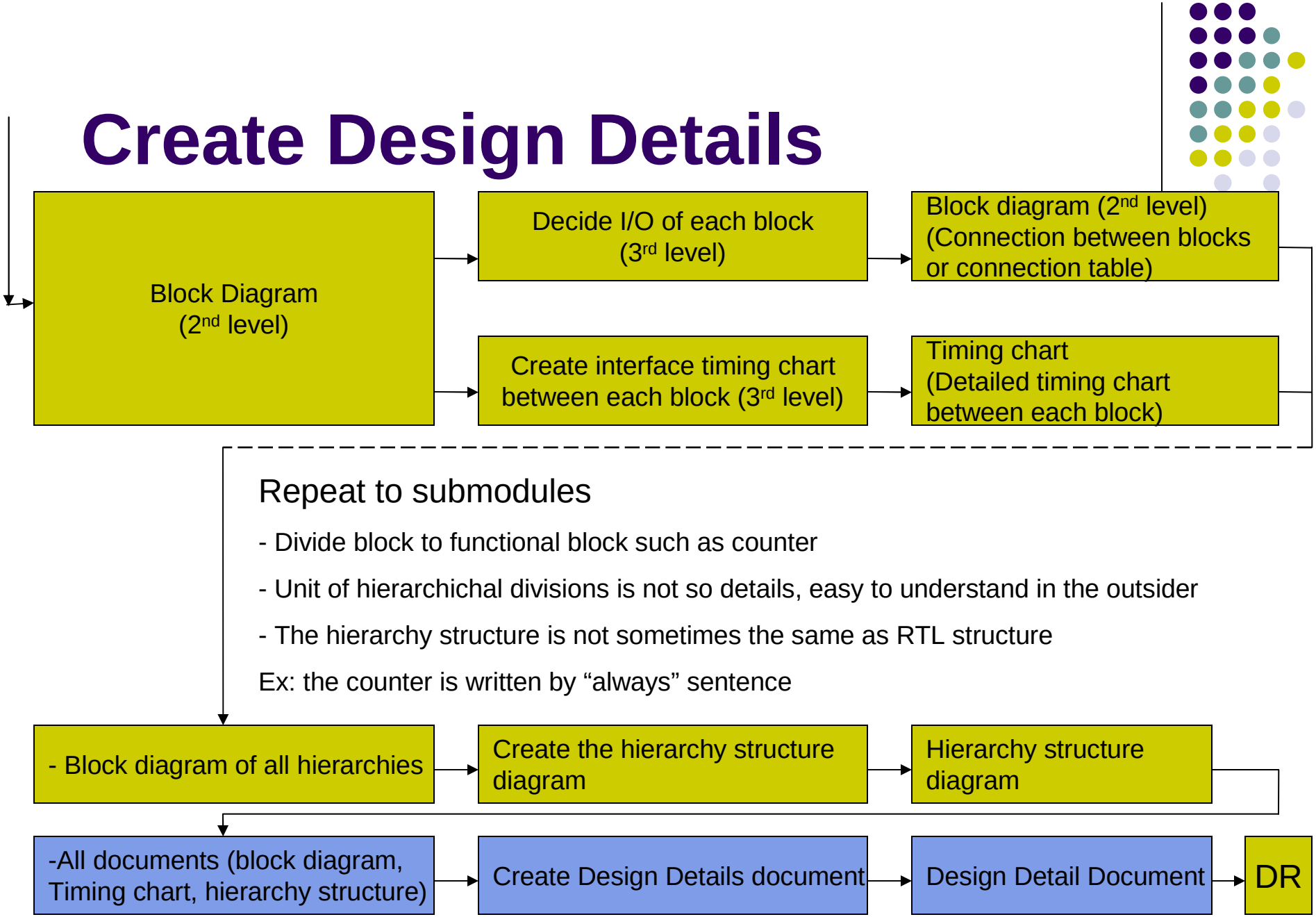
Repeat to submodules

- Divide block to functional block such as counter
- Unit of hierarchichal divisions is not so details, easy to understand in the outsider
- The hierarchy structure is not sometimes the same as RTL structure

Ex: the counter is written by “always” sentence

SEMICON
THINKING OUT OF THE BOX.

Create Design Details

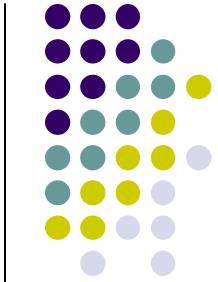


Repeat to submodules

- Divide block to functional block such as counter
- Unit of hierarchichal divisions is not so details, easy to understand in the outsider
- The hierarchy structure is not sometimes the same as RTL structure

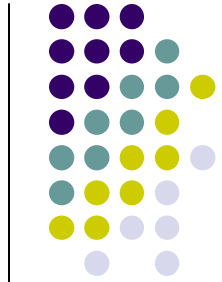
Ex: the counter is written by “always” sentence

Content



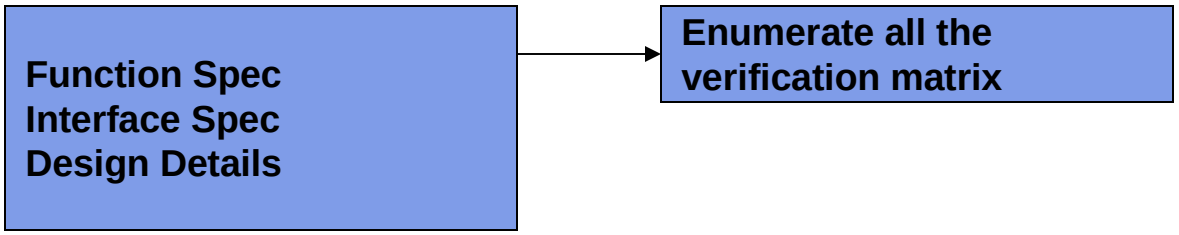
- Design Procedure
- **Verification Plan Procedure**
- Implementation Process

Create Verification Plan

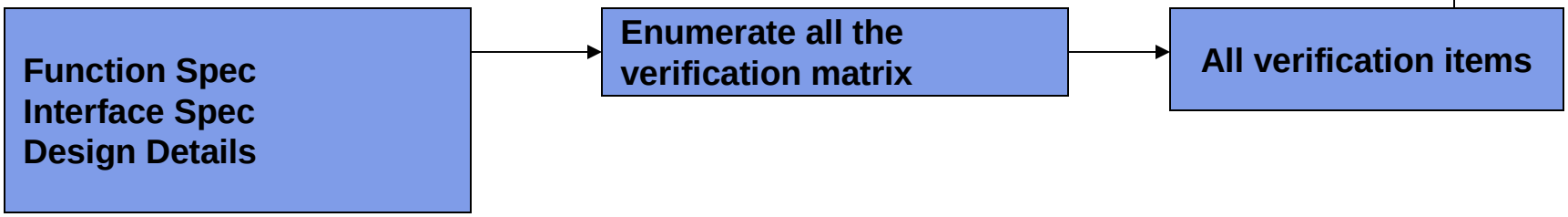


Function Spec
Interface Spec
Design Details

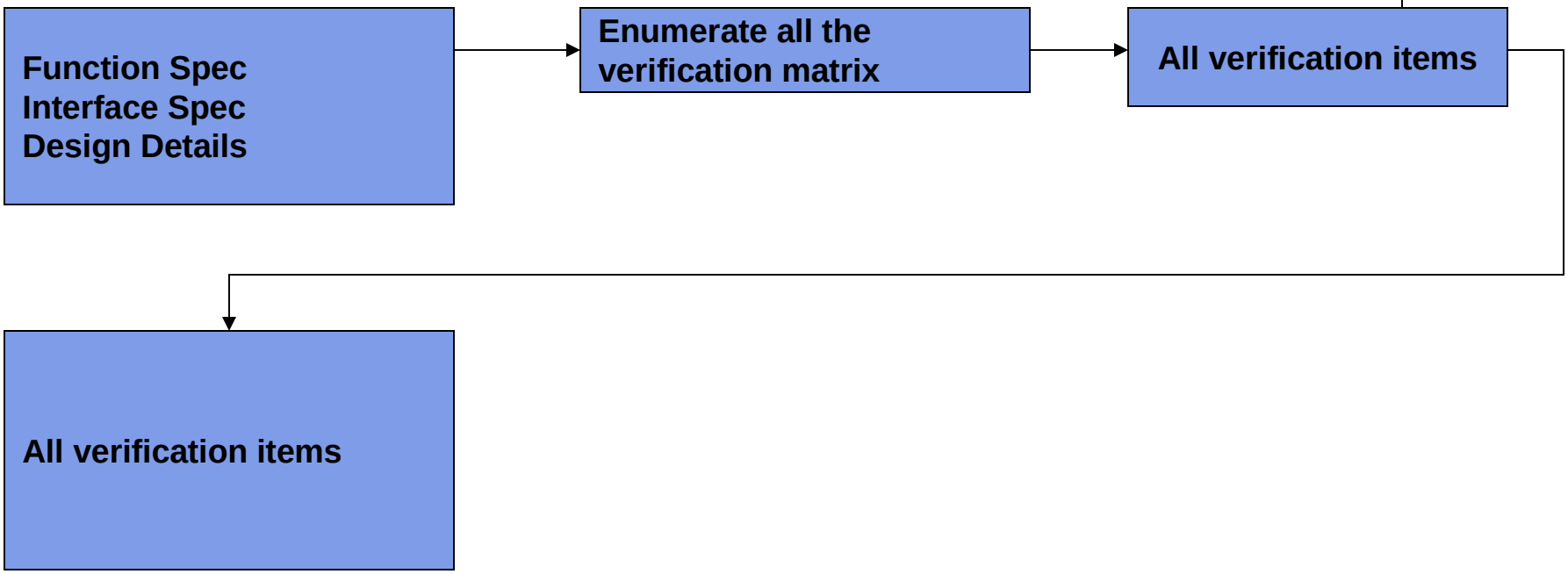
Create Verification Plan



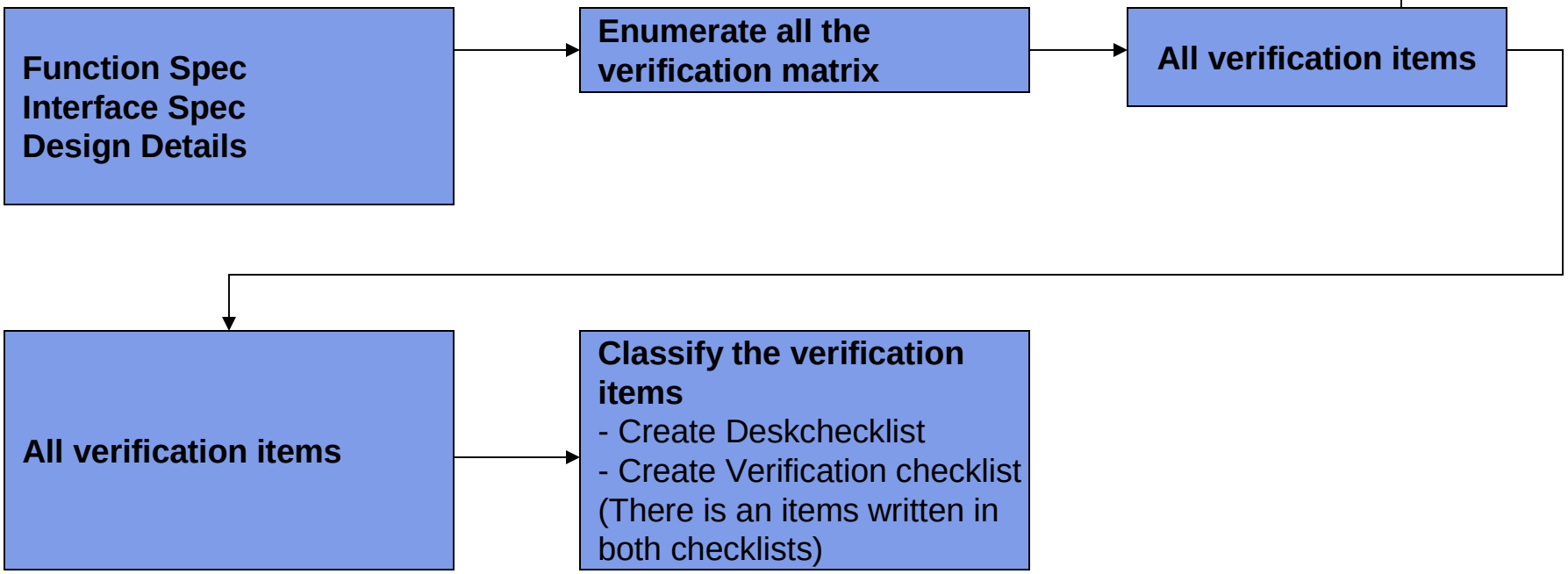
Create Verification Plan



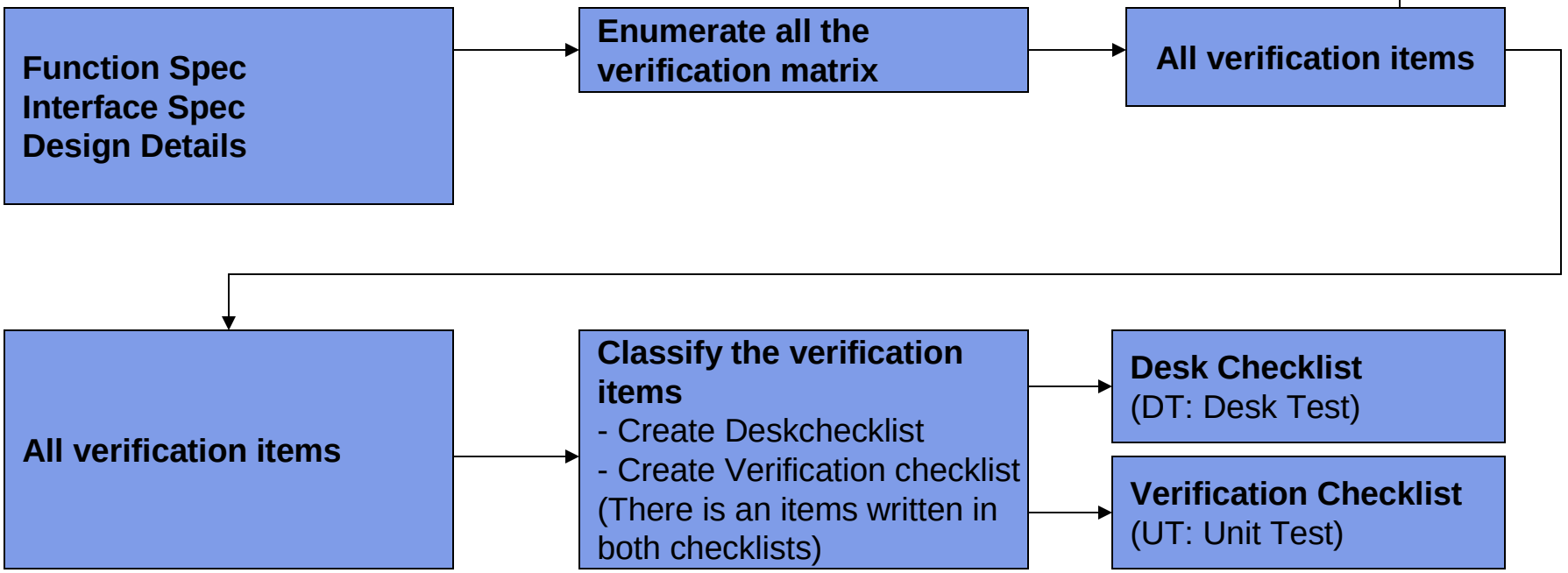
Create Verification Plan



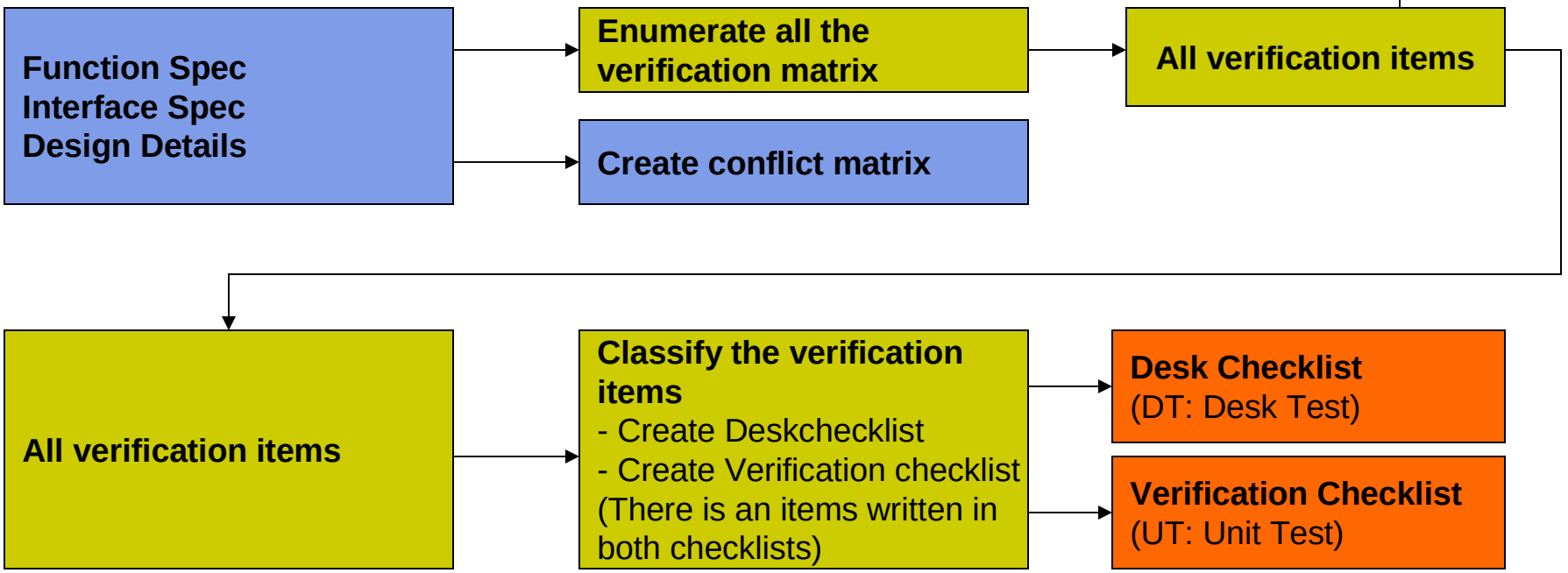
Create Verification Plan



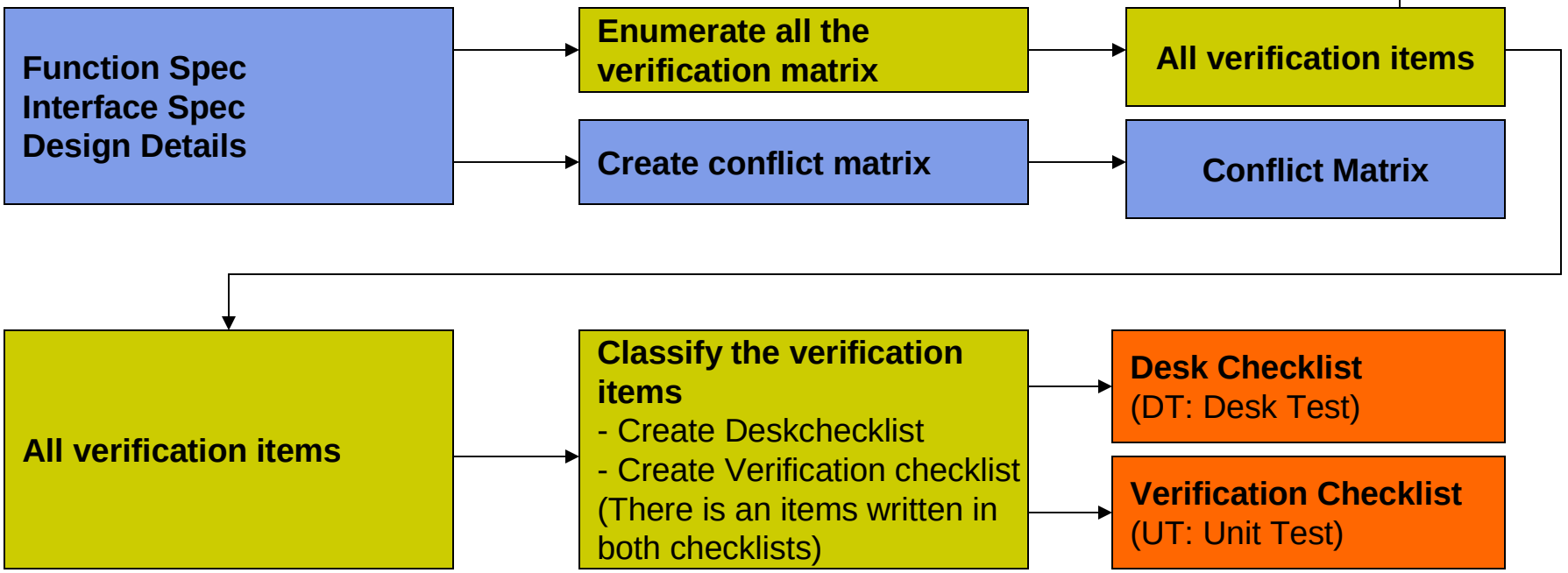
Create Verification Plan



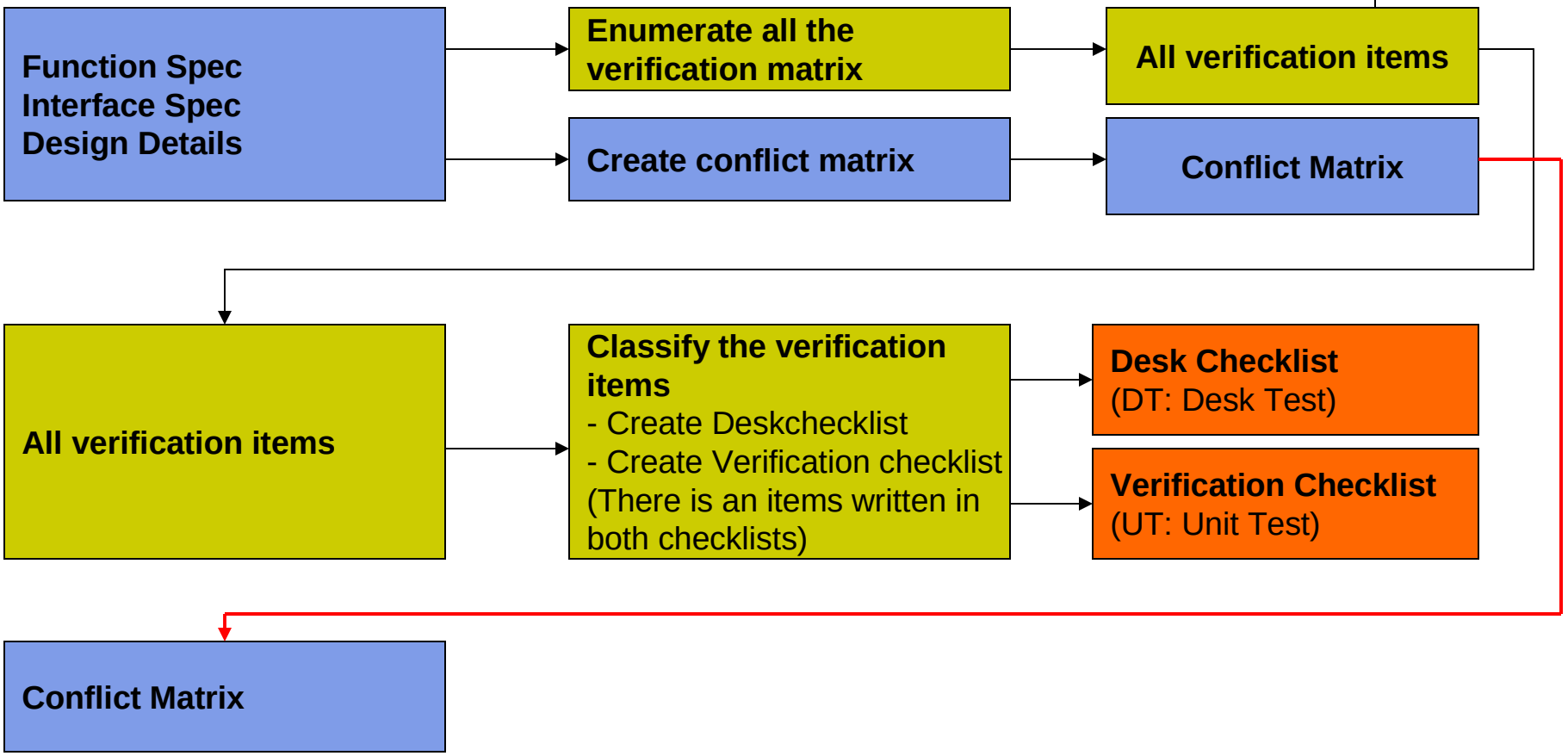
Create Verification Plan



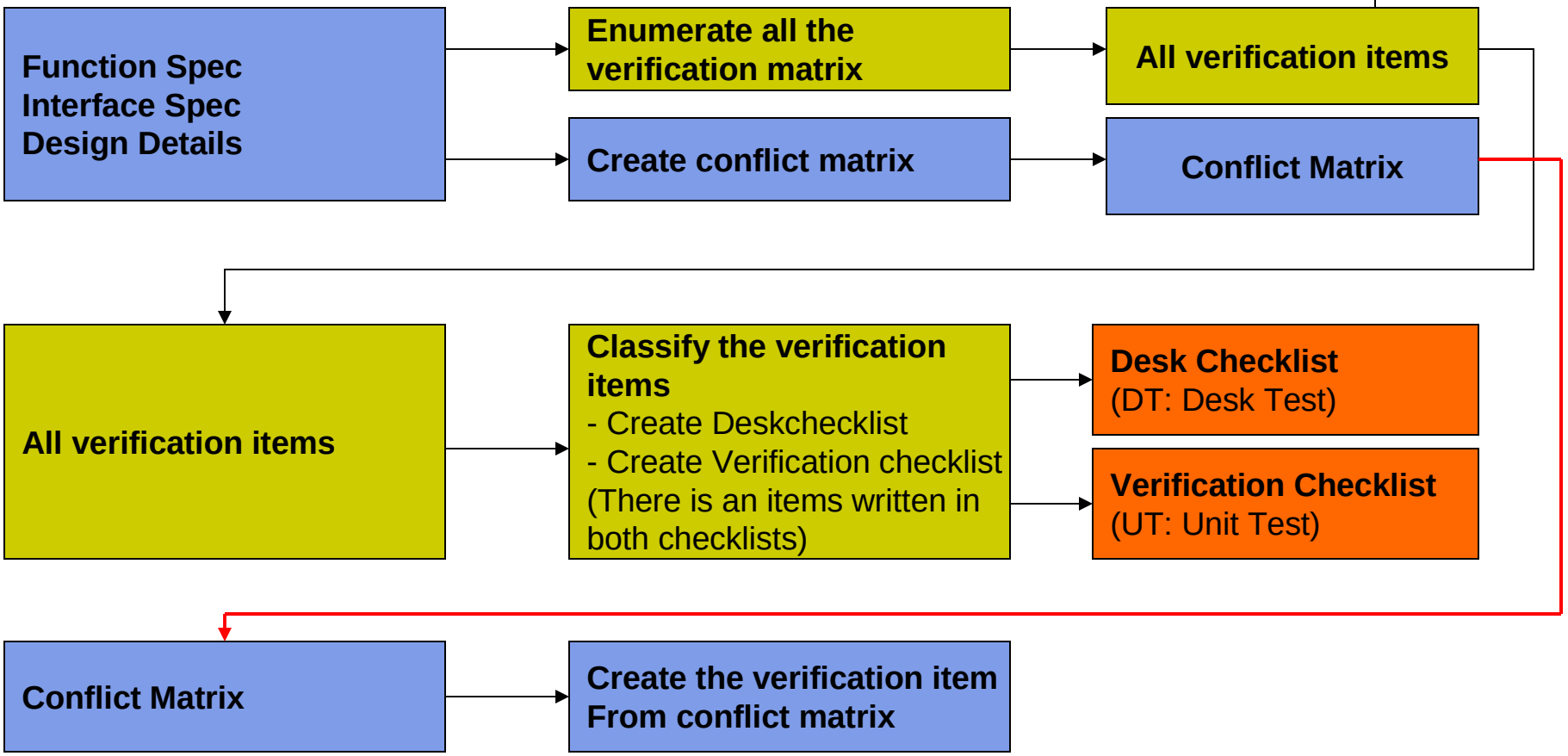
Create Verification Plan



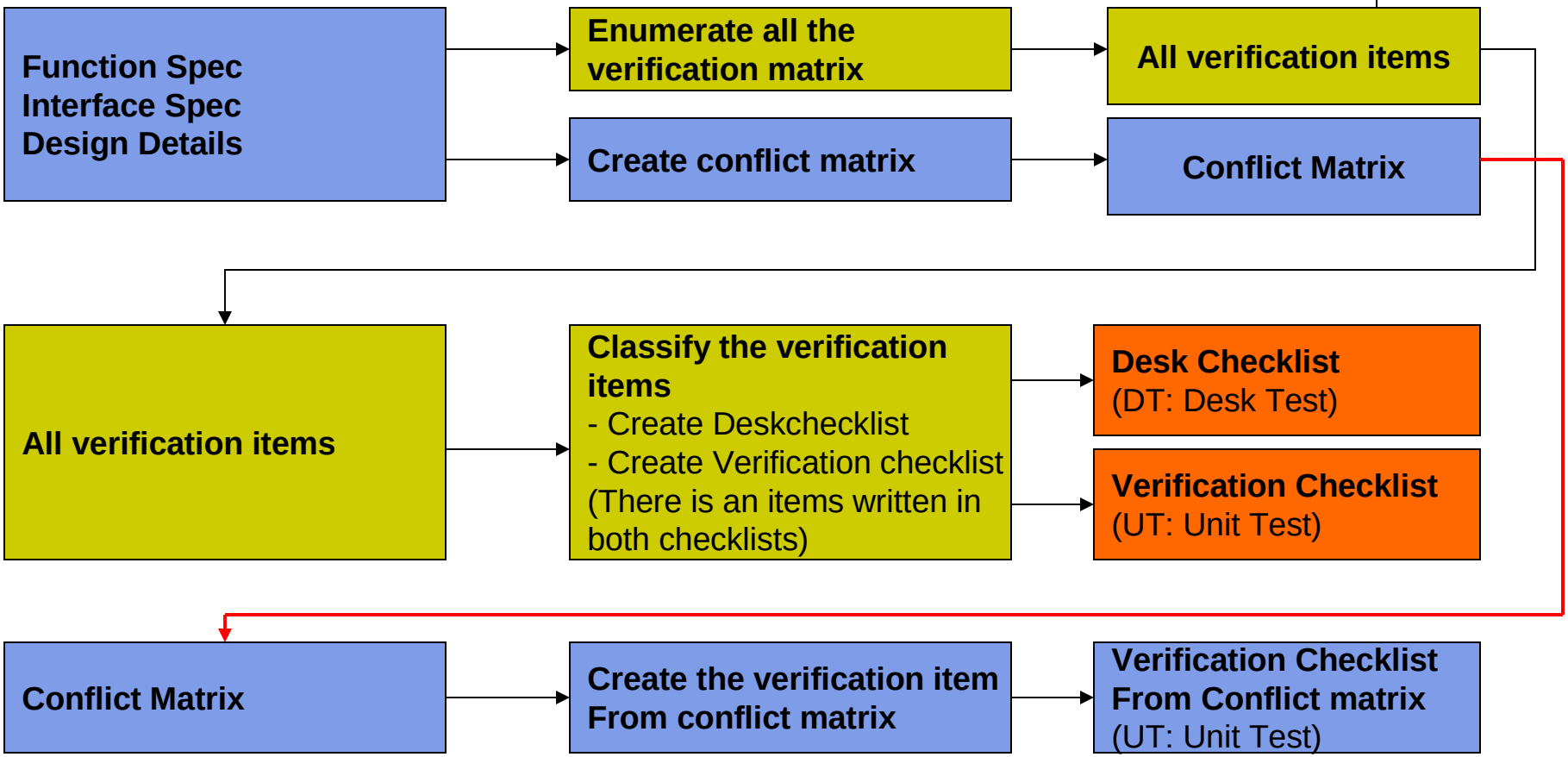
Create Verification Plan



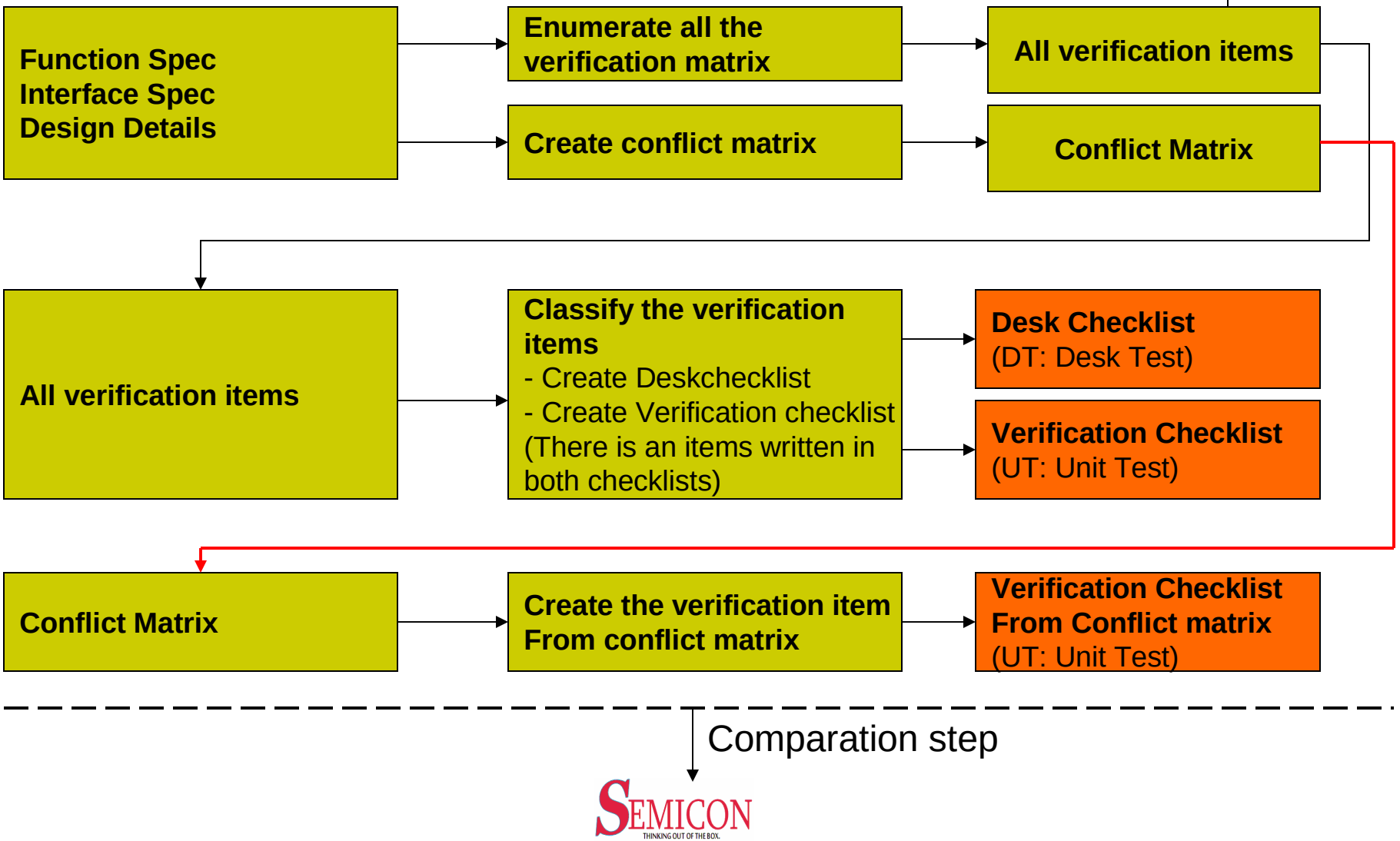
Create Verification Plan



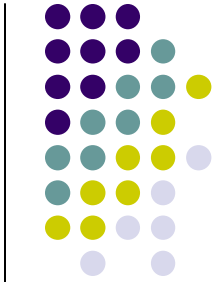
Create Verification Plan



Create Verification Plan

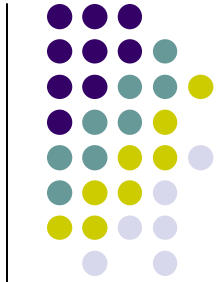


Create Verification Plan



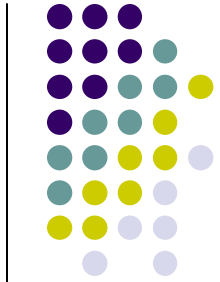
- Whole image of verification plan is understood by planning the function verification plan
- Verification methodology and verification schedule can be estimated
- Enumerate all verification items to understand whole image of the function verification accurately
- Classify the verification items based on the structure, the function of mode, etc.
- Finally, Judge covering verification item by yourself
- However, should be easy to explain the criteria to the outsider

Create Verification Plan



Function Spec
Interface Spec
Design Details

Create Verification Plan



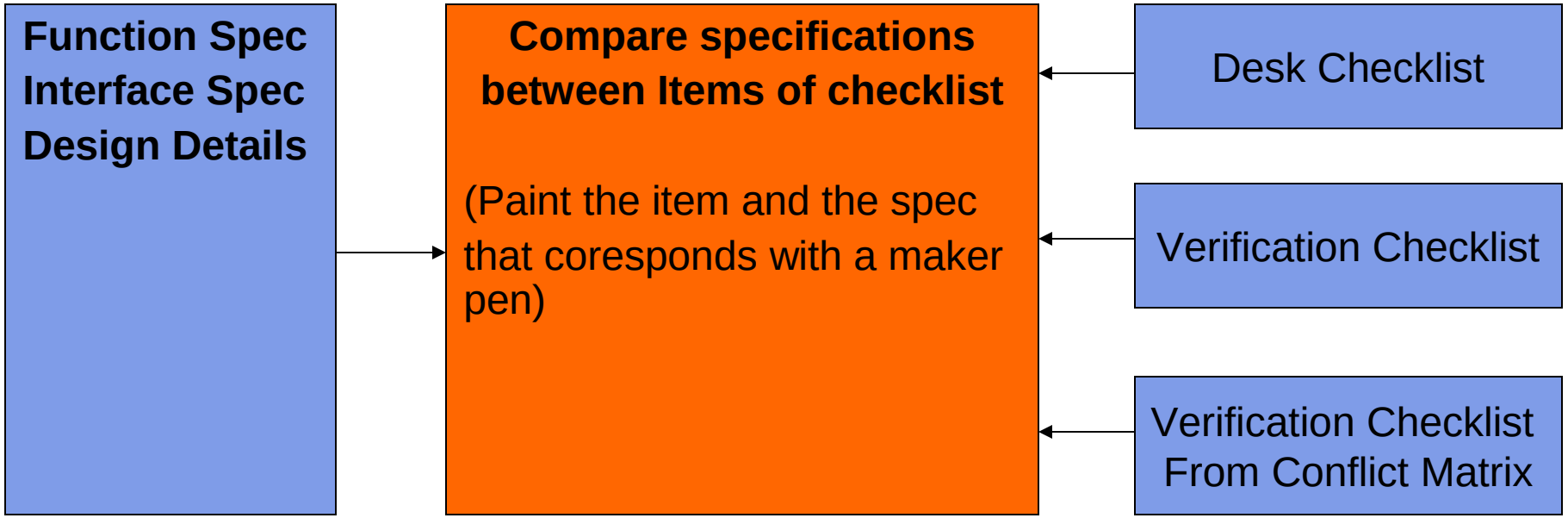
Function Spec
Interface Spec
Design Details

Desk Checklist

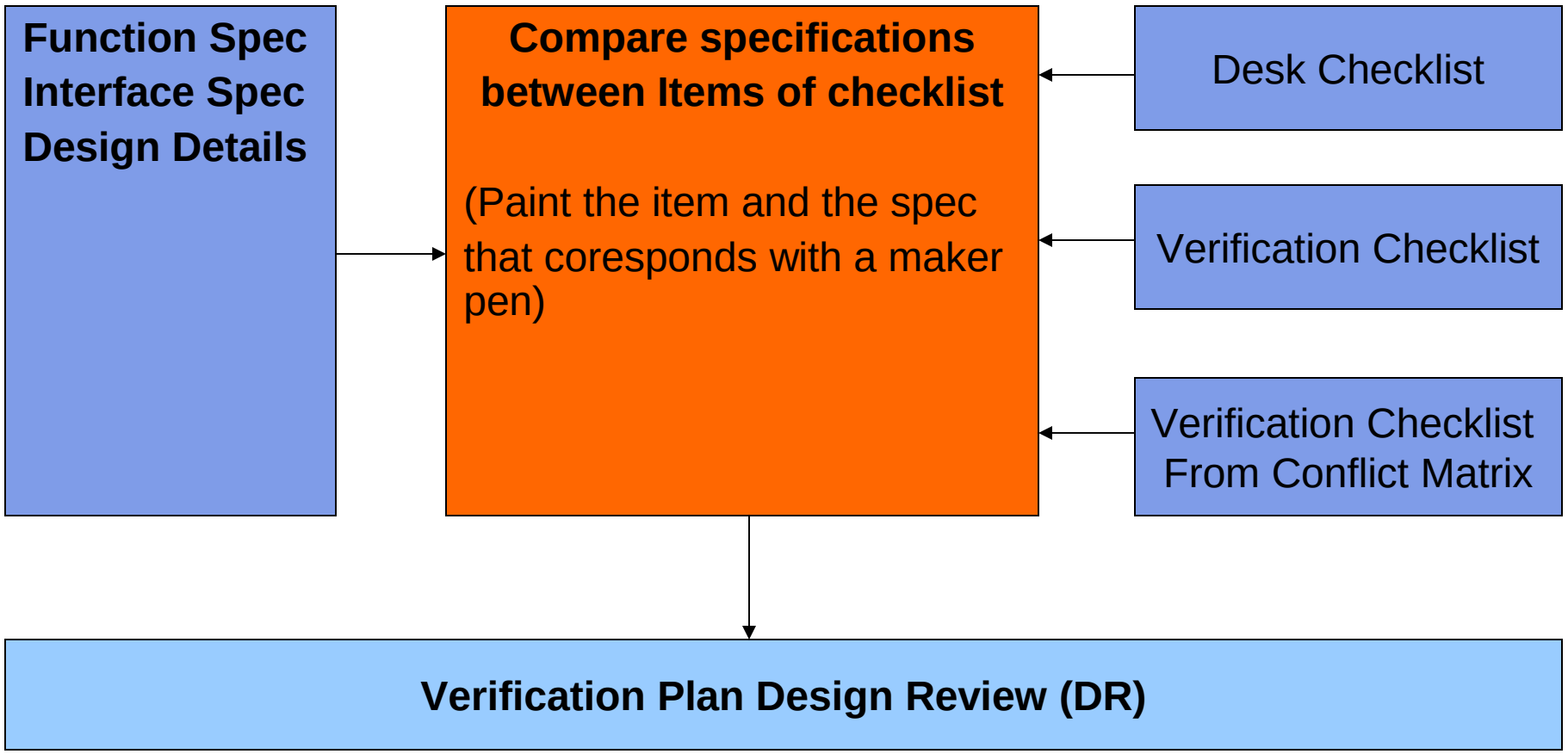
Verification Checklist

Verification Checklist From Conflict Matrix

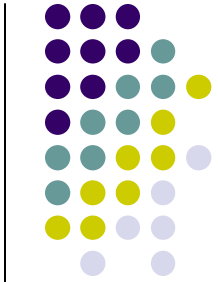
Create Verification Plan



Create Verification Plan

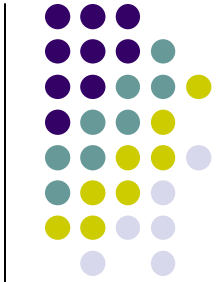


View of enumerating verification items



- Normal function
  - Extract the verification item from the function specification
  - Extract the verification item from CPU interface register
  - Extract the verification item from the modes
- Combination of normal function
  - Extract the verification item that combines the above mentioned normal function
- Continous operation
  - Continuous operation (R -> W -> R) or (R -> R -> R)
- White box verification item
  - Verification item seen from operation boundary condition
  - Verification item seen from conflict condition
  - Verifcation item seen from abnormal operation

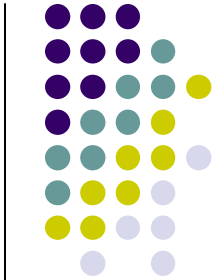
View of Creating Conflict Matrix



- E xtract the states (ST)
  - Combination of values set to control FF = State (Dynamic state)
  - Operatinal mode (Static state)
- Extract the event (EV)
  - Input signal and combination of input signals
  - Assert event and negate event
  - Transaction event such as writing by CPU
- Create the ST & EV matrix
- Create the EV & EV matrix
  - Combination before and behind time of event A and event B
  - Event A and event B occur same time

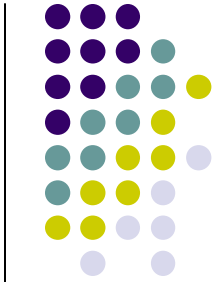
* There are important meanings in looking each cell and the explanation enough of a completed matrix

View of Classifying Verification Item



- Items descibed to Desk Checklist
  - Descibe all function specification
  - Confirm whether all items described in the function specification are written in RTL code
    - Combination of all the thinking fuctions must be confirmed by desk check (Some items are selected and verified by simulation)
    - Difficult items must be confirmed by simulation
- Items descibed to Verification Checklist
  - Items selected from all verification items (Narrow it to a specific condition because it becomes huge in all verification item)
  - Selection method: The item(including explanation) written in the function spec is not enumerated without fail
  - (When verification items become huge, the role of desk check should be reviewed)
  - Describe forgetting neither the test mode nor the concealment function (It is item that is sure to be described in Design Details)

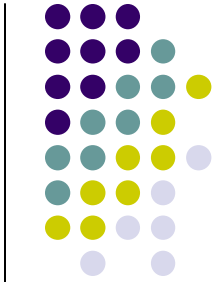
Content



- Design Procedure
- Verification Plan Procedure
- **Implementation Process**

Process

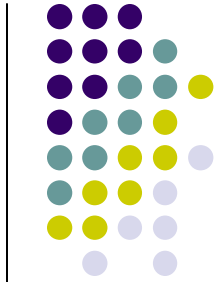
RTL Coding



Process

RTL Coding

TMs Creating



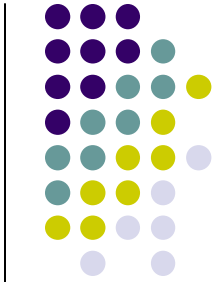
Process

RTL Coding

TMs Creating

Verification by simulation

Process



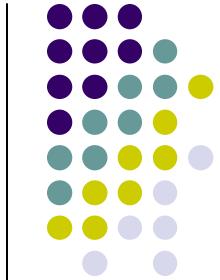
RTL Coding

TMs Creating

Verification by simulation

RTL Coverage

Process



RTL Coding

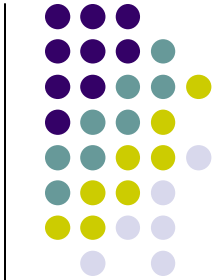
TMs Creating

Verification by simulation

RTL Coverage

Synthesis and
formal verification

Process



RTL Coding

TMs Creating

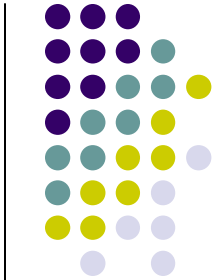
Verification by simulation

RTL Coverage

Synthesis and
formal verification

Checking of Netlist

Process



RTL Coding

TMs Creating

Verification by simulation

RTL Coverage

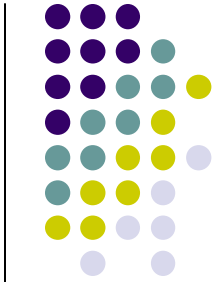
Synthesis and
formal verification

Checking of Netlist

DFT

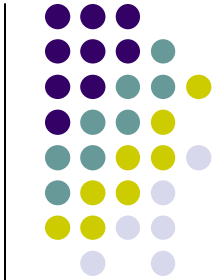
SEMICON
THINKING OUT OF THE BOX.

Work Purpose

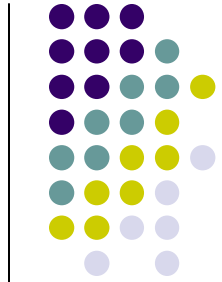


- Understand “Why do you do?” clearly
- Do not misunderstand each work as the purpose
- Clarify the output of each work for the understanding of correct purpose
- Clarify the quality demanded by the output

Outputs



- The output of each work should be a demanded quality
- Consult the project leader when the output is not demanded quality by some reasons
- Report the work result bt the report documents, described result explicity when the output is a demanded quality
- Report on reason anf the background when it is not a demand for quality
- The format of the report is according to it when there is united format



Q & A

SEMICON
THINKING OUT OF THE BOX.