# NuSCR Manual (ver. 1.0)

Dependable Software lab.
KAIST

# Contents

- What is NuSCR?
- Background of NuSCR
- Components of NuSCR
- Variable naming rules
- FOD (Function Overview Diagram)
- Function Variable
- History Variable
- Timed History Variable

KAIST

# What is NuSCR?

- **Nu**clear + **SCR**(Software Cost Reduction)
- Fixed form language for describing requirements
- Suitable for software technology that receives input, performs control logic and gives output
- Suitable for nuclear energy field required technology

KAIST

# Background of NuSCR

- Expansion of the ACEL(Wolsong) method
- ACEL(Wolsong)
  - Basic structure : FOD (Function Overview Diagram)
    - Function : SDT (Structured Decision Table) function table
    - History : State node + function
    - Timing : Timing function
- NuSCR
  - Basic structure : FOD
    - Function : 개선된 SDT function table
    - History : Automata
    - Timing : Time Annotated Automata

KAIST

# Components of NuSCR

- Input variable
- Output variable
- Function variable
- History variable
- Timed history variable
- FOD (Function Overview Diagram)

KAIST

# Variable naming rules

- Add the corresponding prefix to each variable
  - *f_* : function variable
  - *h_* : history variable
  - *th_* : timed history variable
  - *i_* : input variable
  - *o_* : output variable
  - *k_* : predefined constant
  - *g_* : set of function variable, history variable or timed history variable

KAIST

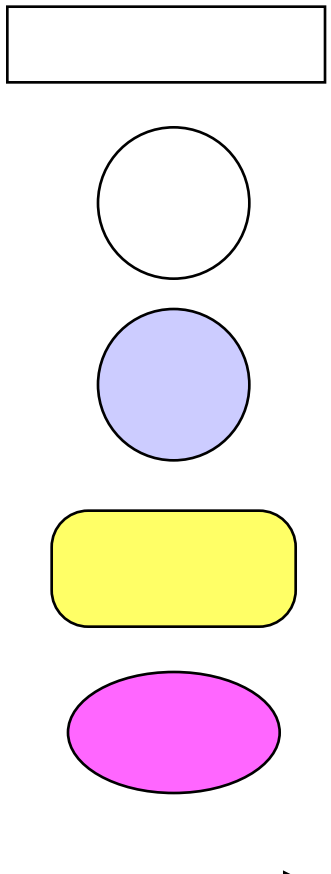
# FOD (Function Overview Diagram)

- A kind of DFD (Data Flow Diagram)
- Describes the relationships between the components of NuSCR
- Display each component with a node
- Display relationships between nodes with one-way arrows
- Use group nodes when composed in classes
- Each node name follows the variable naming rule

KAIST

# Elements represented in FOD

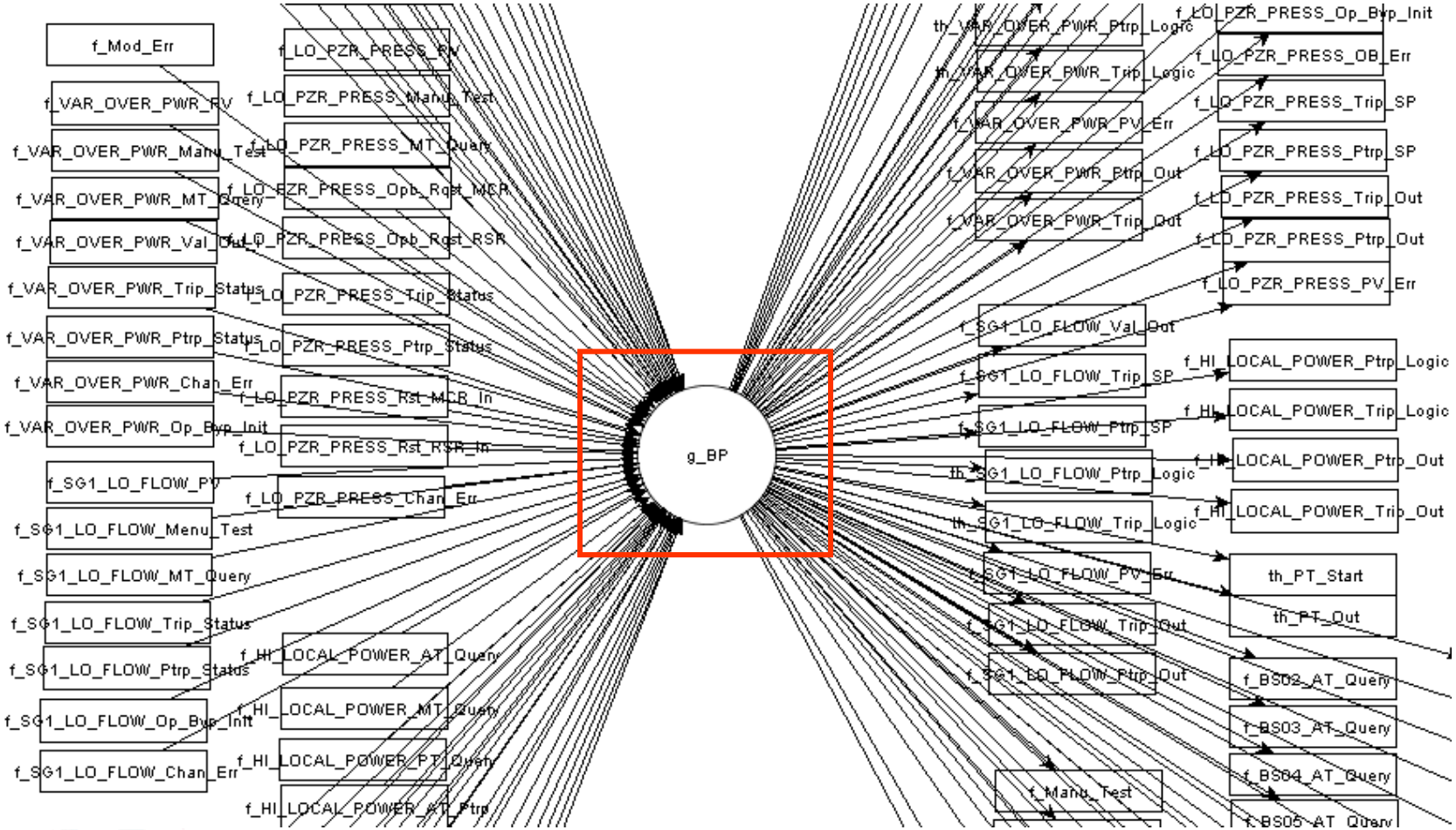
- Input node, Output node
- Group node
- Function node
- History node
- Timed history node
- Data Flow or Transition



KAIST

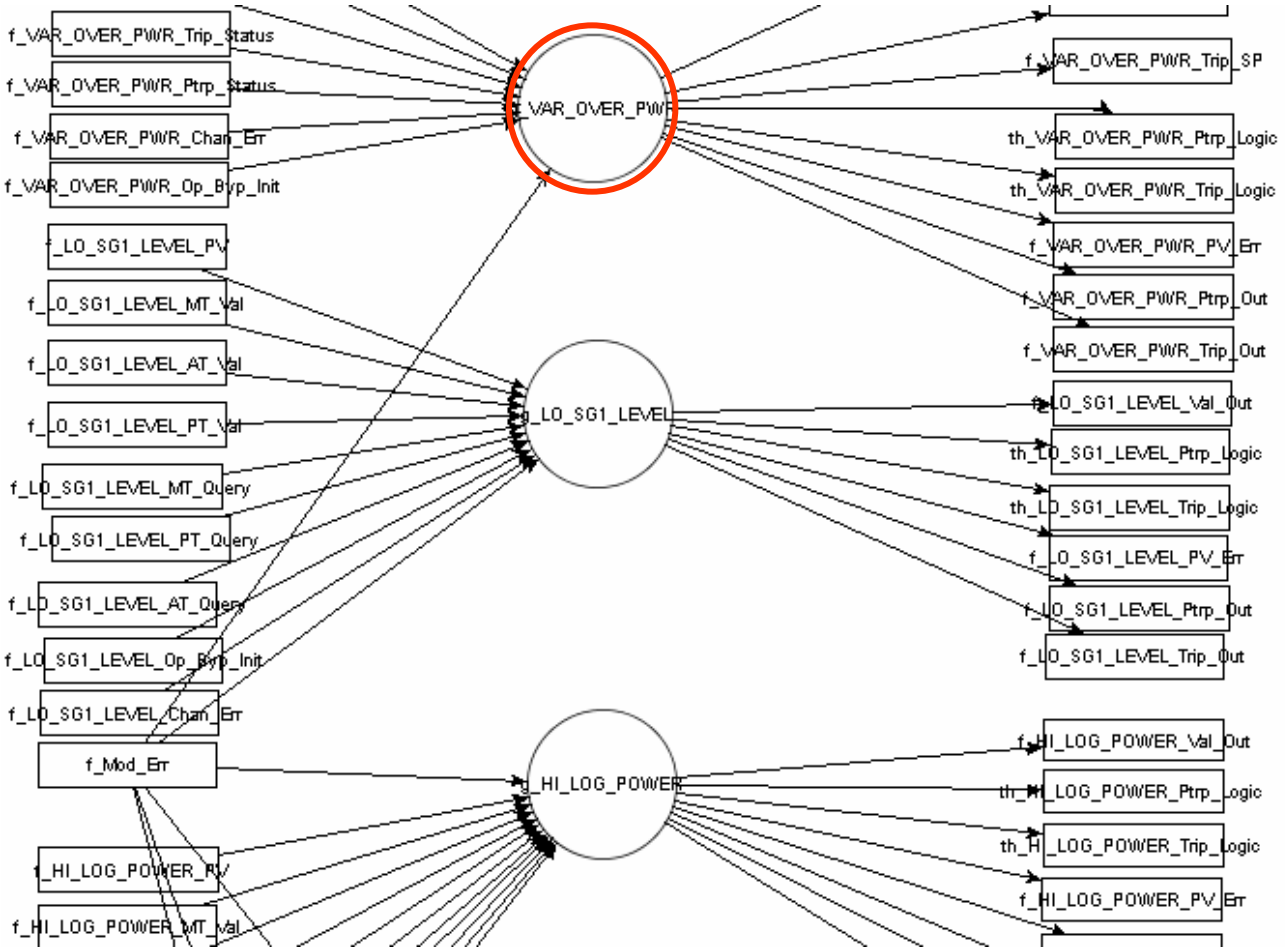
# Example of FOD (1/3)

## g_BP(overview) + External Input/Output



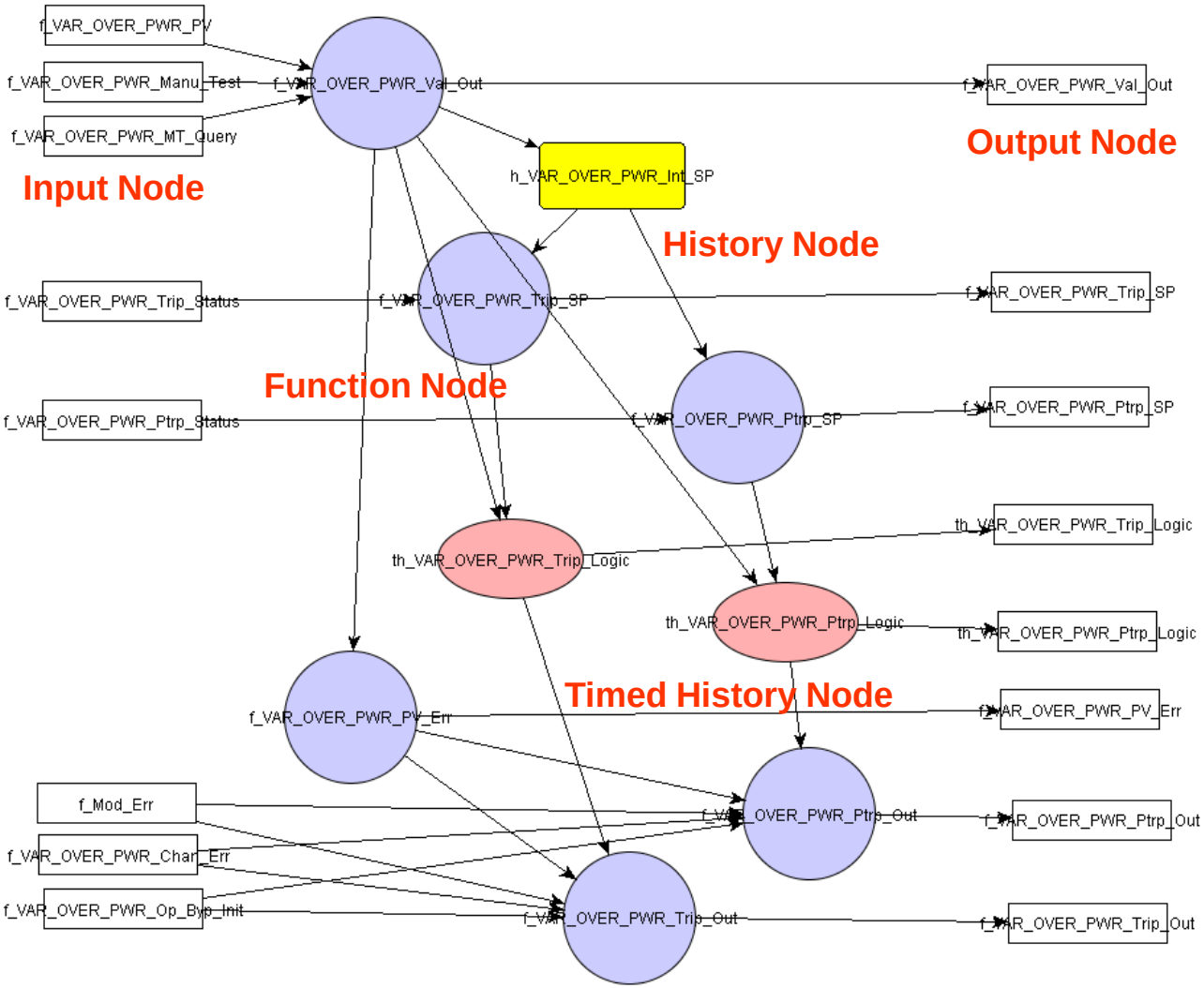
# Example of FOD (2/3)

## g_BP(detailed) + External Input/Output



# Example of FOD (3/3)

## g_BP > g_VAR_OVER_PWR



# Function Variable

- Used to describe the system's functional behavior
- Defined with SDT (Structured Decision Table)
  - SDT is a type of Condition/Action table
  - Once the condition is satisfied, the action is performed
  - Familiar table style for the engineer

KAIST

# SDT (Structured Decision Table)

- Condition
  - Complex condition composed of function variable inputs
  - ie) k_X_MIN <= f_X <= k_X_MAX
- Action
  - Assignments for function variables
  - ie) f_X_Valid := 0

KAIST

# Examples of SDT

| Conditions | 1 | 2 |
|---|---|---|
| k_X_MIN <= f_X <= k_X_MAX | T | F |
| **Actions** | **1** | **2** |
| f_X_Valid := 0 | O | |
| f_X_Valid := 1 | | O |

- SDT defines the function Variable f_X_Valid
- Meaning
  - If f_X is greater than or equal to k_X_MIN, and less than or equal to k_X_MAX (condition),
  - Assign 0 to f_X_Valid (action)

KAIST

# Example of SDT from RPS items

- Example of function variables defined through SDT

f_LO_SG1_LEVEL_PV_Err

**Structured Decision Table:**

| Conditions | 1 | 2 | 3 |
|---|---|---|---|
| f_LO_SG1_LEVEL_Val_Out > k_LO_SG1_LEVEL_PV... | T | - | F |
| f_LO_SG1_LEVEL_Val_Out < k_LO_SG1_LEVEL_PV... | - | T | F |
| | | | |
| **Action** | **1** | **2** | **3** |
| f_LO_SG1_LEVEL_PV_Err := true | O | O | |
| f_LO_SG1_LEVEL_PV_Err := false | | | O |
| | | | |

# History Variable

- Used to describe system's condition based action
- Defined with a FSM (Finite State Machine)
  - Components of FSM
    - Finite number of states
    - Transitions between states

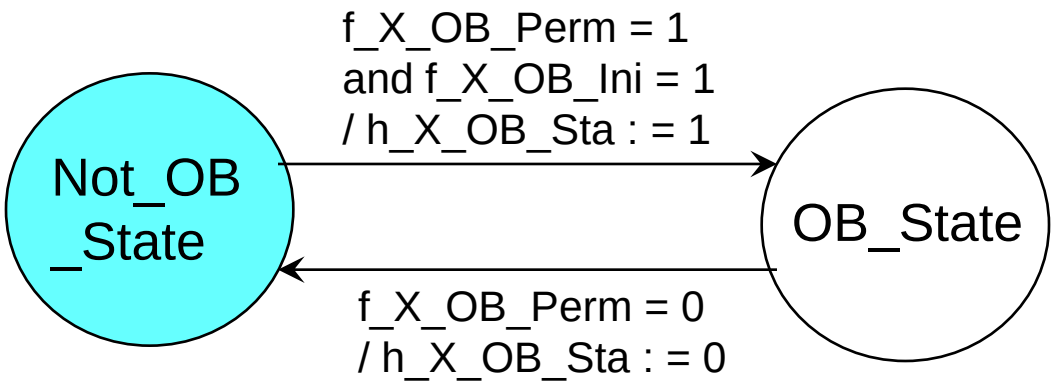
KAIST

# FSM (Finite State Machine)

- State
  - Express each of the system's states
  - ie) A switch has two states : On and Off
- Transition
  - Represents the changes between states
  - Expressed with arrows
  - Each transition has a label
  - label form → Conditions/Actions

KAIST

# Example of FSM (Finite State Machine)



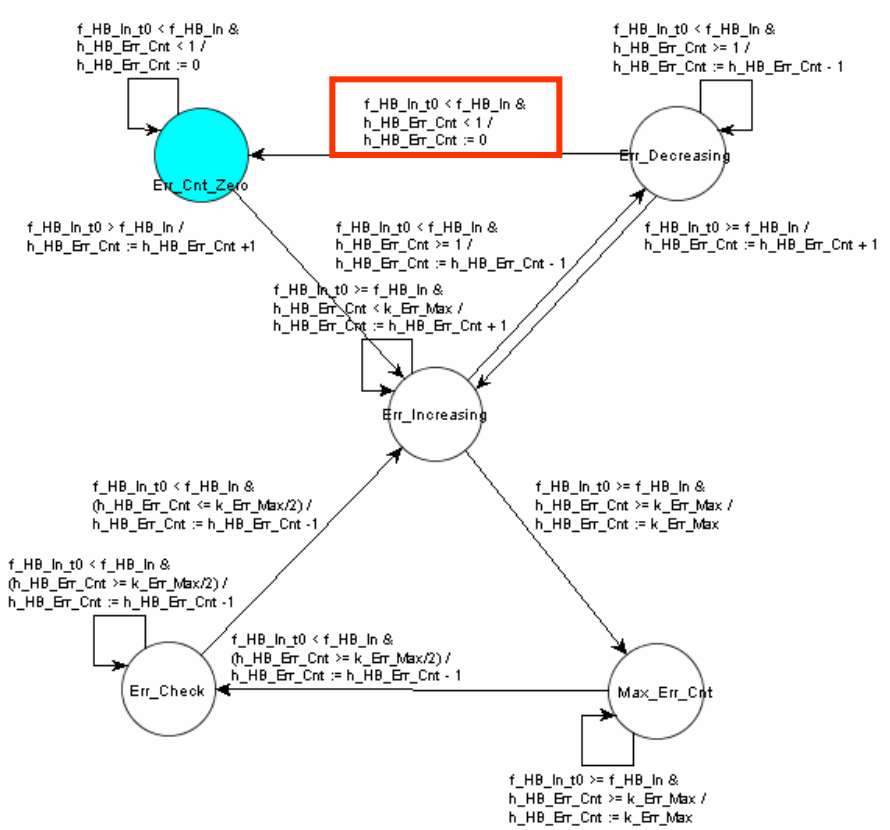

- FSM that defines the history variable h_X_OB_Sta
- Meaning
  - In the initial state NOT_OB_STATE
  - If the conditions f_X_OB_Perm = 1 and f_X_OB_Ini = 1 are satisfied (condition)
  - Assign the value 1 to h_X_OB_Sta (action)
  - Move to the OB_State (transition)

KAIST

# Example of FSM from RPS items

- Example of history variables defined through FSM

h_HB_Err_Cnt




- **Condition** : f_HB_In_t0 < f_HB_In & h_HB_Err_Cnt < 1
- **Action** : h_HB_Err_Cnt := 0

# Timed History Variable

- Used to describe system's time related actions
- Defined with TTS (Timed Transition System)
  - TTS is an extension of FSM
  - Time Annotated Automata
  - Adds a time restriction to FSM's transition condition
  - Attaches a time restriction in the form of [a,b] in front of the condition

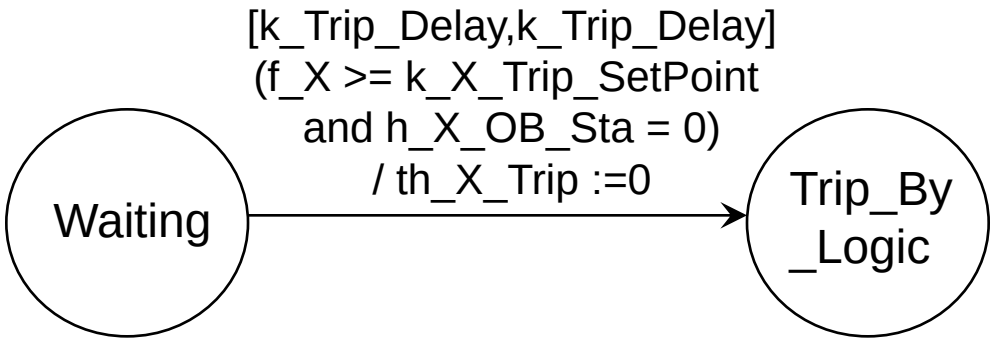
KAIST

# TTS (Timed Transition System)

- State
  - Describes the systems' different states
- Transition
  - Represents the changes between states
  - Expressed with arrows
  - Every transition has a label
  - label format → [$Time_1$,$Time_2$]Conditions/Actions
  - ie) [1,4]condition=0/action:=1
    - If the condition=0 is maintained for a term of 1~4 hours, assign action=1 and change state

KAIST

# Example of TTS (Timed Transition System)



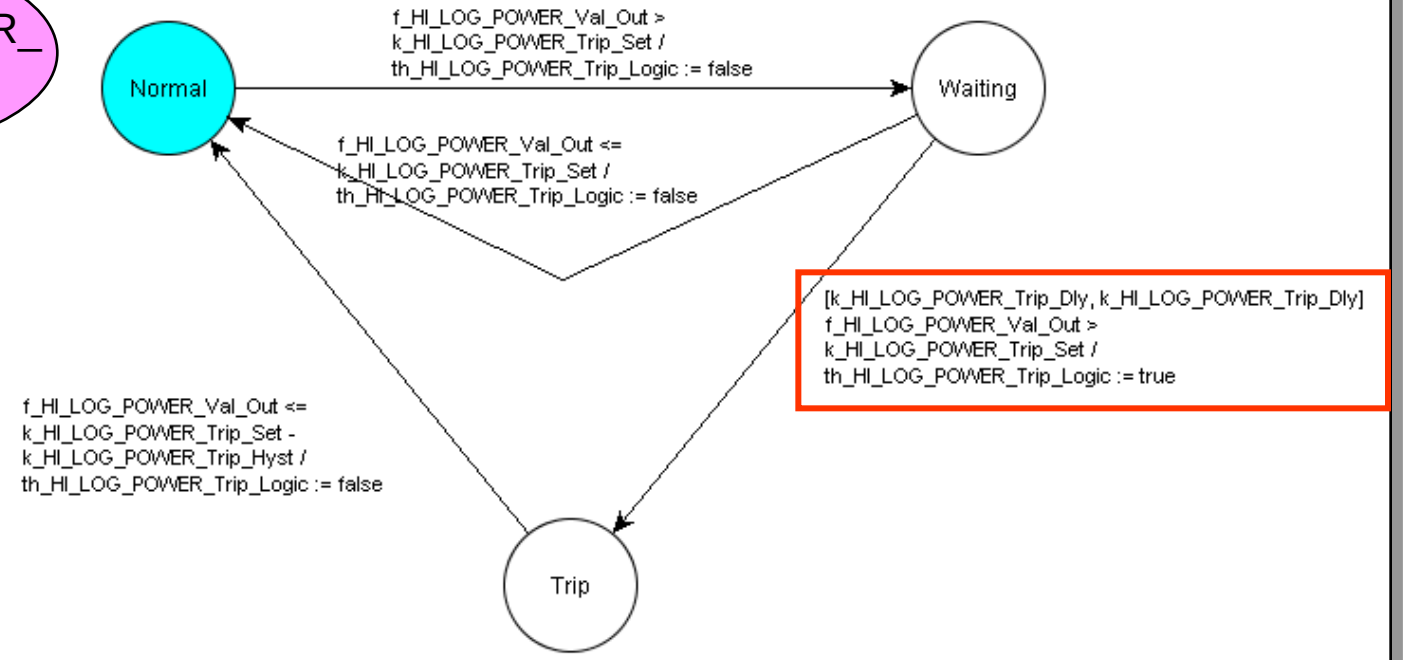

- TTS that defines a part of Timed History Variable th_X_Trip
- Meaning
  - In Waiting state
  - For k_Trip_Delay hours (Time Limit)
  - If f_X >= k_X_Trip_SetPoint and h_X_OB_Sta = 0 conditions are satisfied and maintained (condition)
  - Assign th_X_Trip the value 0 (action)
  - Move to the Trip_By_Logic state (transition)

KAIST

# Example of TTS from RPS items

- Example of Timed History Variable defined through TTS

th_HI_LOG_PWR_Trip_Logic




- **Time duration** : [k_HI_LOG_POWER_Trip_Dly, k_HI_LOG_POWER_Trip_Dly]
- **Condition** : f_HI_LOG_POWER_Val_Out > k_HI_LOG_PWR_Trip_Set
- **Action** : th_HI_LOG_PWR_Trip_Logic := true

KAIST