**Test Case Generator**

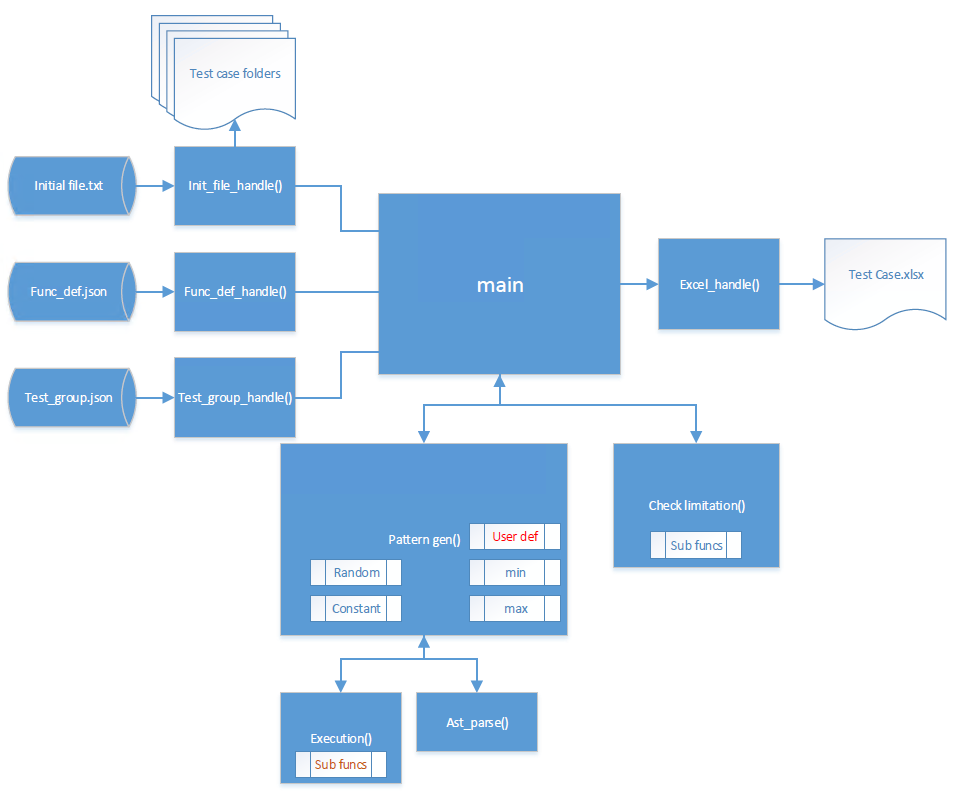
**User Guide**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Revision** | **Editor** | **Description** |
| 2018/01/19 | 0.01 | EasanJhang | Initial draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. **Case Gen Tool Block Diagram**

The case gen tool is programmed by python. The block diagram is shown below. Users need to provide three files to the tool. Include the initial file (.txt), function definition (.json) and test group info (.json), and its output contains the test case folders and test\_case.xlsx.



pattern\_gen() 的設計想法，參考以往excel gen case的方式，每個register一樣會存在獨立性與相依性的問題，

* 假設有 [reg1, reg2, reg3, reg4 ] 4個register，相依關係為 **reg1** => **reg2** => **reg3** => **reg4**
* 呈上，**reg4** 為最 independent，因此code gen會由 **reg4** 開始計算數值，順序為 **reg4** -> **reg3** -> **reg2** -> **reg1**
* 假如 獨立性與相依性 關係有**遞迴關係**的話( **reg1** => **reg2** => **reg3** => **reg4** => **reg1** )，可能就必須用到 user\_define，不過目前 user\_define 部分**尚未**設計框架。
* 上圖，**Sub funcs**部分為目前case gen tool自定義給FuncDef作為expression使用的API，這部分需依需求陸續新增功能。

1. **The in/output file list**

|  |  |  |
| --- | --- | --- |
| **Input** | Initial\_file.txt | Provide default and bit number information |
| Func\_def.json | provides definition of the expression and constraints for all the registers |
| Test\_group.json | Provide the test groups definition |
| **output** | Test\_Case.xlsx | Output report of all test patterns information to excel file |
| Test Case folders / TestCase.txt | Output all the test patterns to the text files |

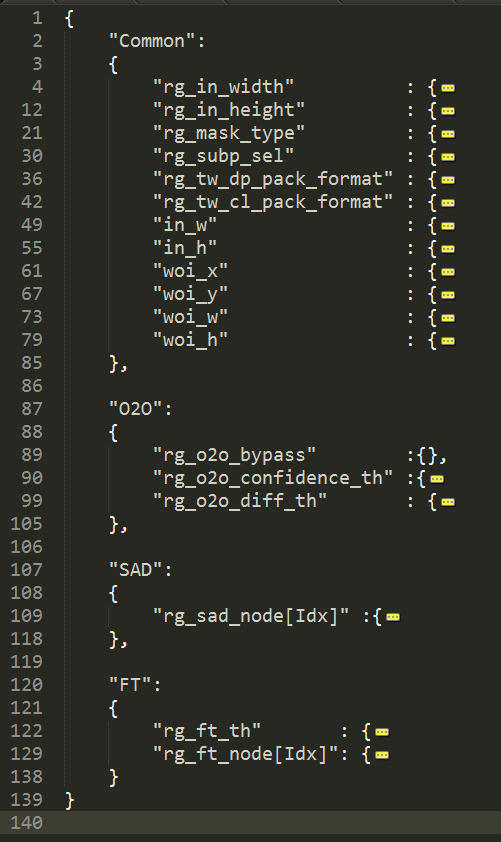
1. Json File Format

相關Sample code可參考 **[SVN]:** ASICSW\_MiniISP\_D3S\trunk\Tool\pyCaseGenTool\

1. Function define (.json)

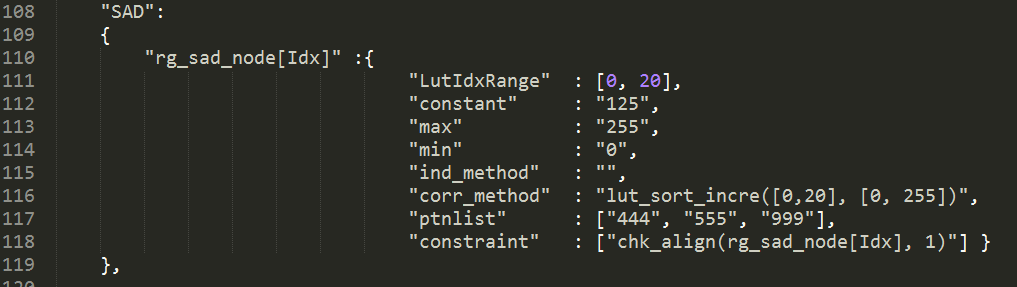
* Sample:

FuncDef name



Register list

* **Func def name:** 在function define .json裡，第一層為自訂的function name
* **Register list:** 在function name裡，第二層為register list，每一個module在分群時，建議將該function所有有相依性的register放在同一群的function define。
  + 目前register list support兩種形式，(1). Normal register (2). LUT Table register
    1. Normal register: 請直接填register name
    2. LUT Table register: 格式為reg\_key\_word**[Idx]**，並且在**group info**記得填上"LutIdxRange"



* **Register info:** 將每一個register展開後，第三層為register info，該層將定義每個register預期要代入code gen case的計算式。而這邊預期將每項attribute均以string定義。



Register info.

Note: 在Test group (.json) 中會定義每個Test group預期要產生pattern的條件選項，而這些選項包含: {"constant", "max", "min", "random", "default", "user\_define"}

|  |  |
| --- | --- |
| **Register info attribute** | **Description** |
| **"constant"** | When the test group definition is "constant", the case gen will execute this expression |
| **"max"** | When the test group definition is "max", the case gen will execute this expression |
| **"min"** | When the test group definition is "min", the case gen will execute this expression |
| **"ind\_method"** | When the test group definition is "random", the case gen will execute this expression first. |
| **"corr\_method"** | When the test group definition is "random", the case gen will execute this expression after the independent method is executed. |
| **"user\_define"** | undefined |
| **"constraint"** | 這項attribute會在check limitation時，作為檢查每個register的HW limitation使用 |

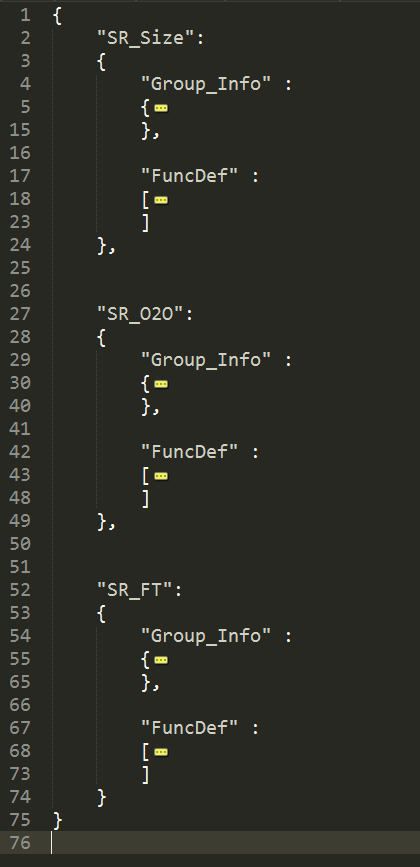
* 在.json裡，我們會將每項**Register info attribute**當作dictionary來使用，而目前開放的有"constant", "max", "min", "ind\_method", "corr\_method", "constraint"這幾種key可以填寫。
* 呈上，每個key裡面會對應到一組**value**，而這些**value**的填寫方式，必須以string的格式來填寫。
* **Value**的內容分為兩種情況，(1). 一般的register，及(2). LUT table register
  1. 一般的register，這些**value**可以是一段expression也可以是一個constant，不過有一點要注意，如果需要用到python的library api時，目前這邊會有限制，僅能使用return為Int的自定義API，如下(表1)。
  2. 假如該register為LUT table，目前僅能填一個constant，或是case gen tool開放的return為list的自定義API，如下(表2)。
* Attribute為 **"ind\_method"**, **"corr\_method"**時，是當Test group的FuncDef選擇**"Random"** 時被使用，因register會有獨立關係與相依關係，因此這兩個屬性會拿來特殊應用
  1. 第一Run程式會先去判斷**"ind\_method"**是否有值，若有值，**"ind\_method"**會首先被執行
  2. 第二Run之後，程式會執行**"corr\_method"**，並檢查expression裡是否有尚未更新的register，若expression裡頭的register均已被更新，則會執行這段expression，反之則留到下一Run執行。
  3. 因此建議使用者在填**"ind\_method"**, **"corr\_method"**這兩個expression，請擇一填寫，避免誤判。
* Attribute為 **"constraint"**，**Value**需填寫expression計算結果為bool的式子，若欲使用Tool開放的自定義API可參考下(表3)。
* 關於Case gen tool自定義的API，可參考 ./lib/execution.py, ./lib/exec\_subfunc.py

|  |  |
| --- | --- |
| **Normal register (return int)** (表1) | |
| **API name** | **Description** |
| i\_max(arg1, arg2, \*args) | return **maximum** |
| i\_min(arg1, arg2, \*args) | return **minimum** |
| i\_randint(a\_start, a\_end) | return a\_start<= **(random)** <= a\_end |
| i\_randlist(a\_list) | return list(**random**) |
| i\_rand\_align(a\_start, a\_end, a\_align) | return a\_start<= align**(random)** <= a\_end |
| i\_rand\_even(a\_start, a\_end) | return a\_start<= **even** <= a\_end |
| i\_rand\_odd(a\_start, a\_end) | return a\_start<= **odd** <= a\_end |
| i\_max\_bits(a\_bitnum) | return **2(max bit)** |
| i\_align\_ceil(a\_val, a\_align) | return **ceil(align(random))** |
| i\_align\_floor(a\_val, a\_align) | return floor**(align(random))** |

|  |  |
| --- | --- |
| **LUT Table register (return list)** (表2) | |
| **API name** | **Description** |
| lut\_sort\_incre(a\_idx\_range, a\_val\_range) | return a random and increase list  the index range is **a\_idx\_range**  the value ranger is **a\_val\_range** |
| lut\_sort\_desc(a\_idx\_range, a\_val\_range) | return a random and decrease list  the index range is **a\_idx\_range**  the value ranger is **a\_val\_range** |
| lut\_sort\_incre\_align(a\_idx\_range, a\_val\_range, a\_align) | return a random, increase and align list  the index range is **a\_idx\_range**  the value ranger is **a\_val\_range** |
| lut\_sort\_desc\_align(a\_idx\_range, a\_val\_range, a\_align) | return a random, decrease and align list  the index range is **a\_idx\_range**  the value ranger is **a\_val\_range** |
| lut\_rand(a\_idx\_range, a\_val\_range) | return a random list  the index range is **a\_idx\_range**  the value ranger is **a\_val\_range** |
| lut\_rand\_align(a\_idx\_range, a\_val\_range, a\_align) | return a random and align list  the index range is **a\_idx\_range**  the value ranger is **a\_val\_range** |

|  |  |
| --- | --- |
| **constraint (return bool)** (表3) | |
| **API name** | **Description** |
| chk\_align(a\_reg, a\_align) | Check the register(**a\_reg**) is align(**a\_align**) |
| chk\_even(a\_reg) | Check the register(**a\_reg**) is even(**a\_align**) |
| chk\_odd(a\_reg) | Check the register(**a\_reg**) is odd(**a\_align**) |

1. Test group (.json)



Group info

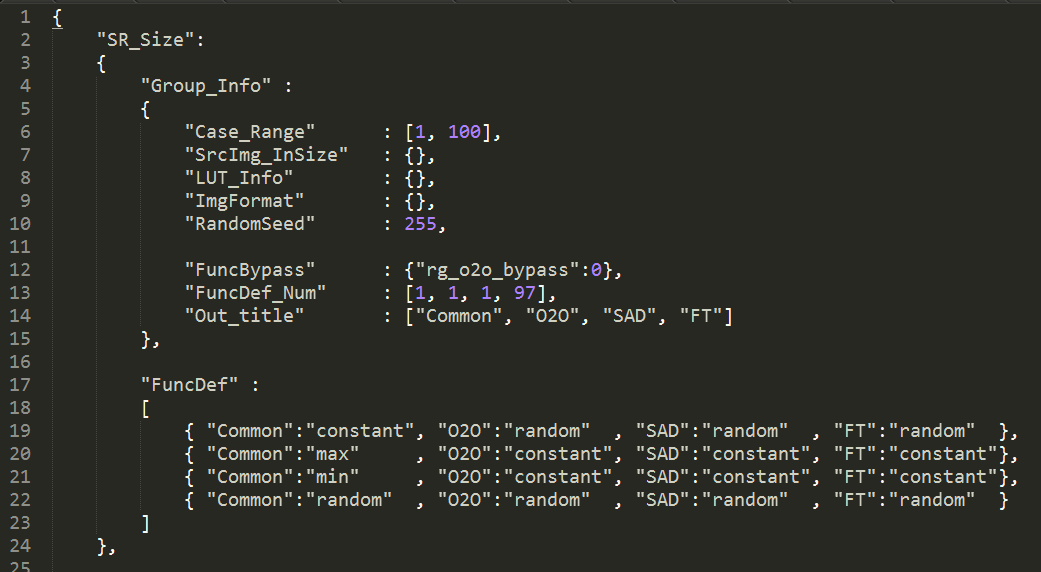
Function define

Test Group name

* **Test Group name:** 在Test group .json中，第一層為自訂的test group name，
* **Group info:** 在自定義的group name中，第二層分別有group info以及FuncDef，分別提供與該Test group需要的資訊。

|  |  |
| --- | --- |
| **Group info attribute** | **Description** |
| **"Case\_Range"** | Define the range of the case index |
| **"SrcImg\_InSize"** | undefined |
| **"LUT\_Info"** | undefined |
| **"ImgFormat"** | undefined |
| **"RandomSeed"** | Provide random seed for case gen reference |
| **"FuncBypass"** | Define bypass register to control the specified function. |
| **"FuncDef\_Num"** | Define the number of cases for each function definition. |
| **"Out\_title"** | Define the output function title for test group sheet. (Excel file) |

* **Function define:** 提供預期要產生pattern的function定義，而該function name必須與function define .json定義的一致。且每個function可以有以下 {"constant", "max", "min", "random", "default", "user\_define"} 這幾種組合。



1. **Code Gen Flow**

