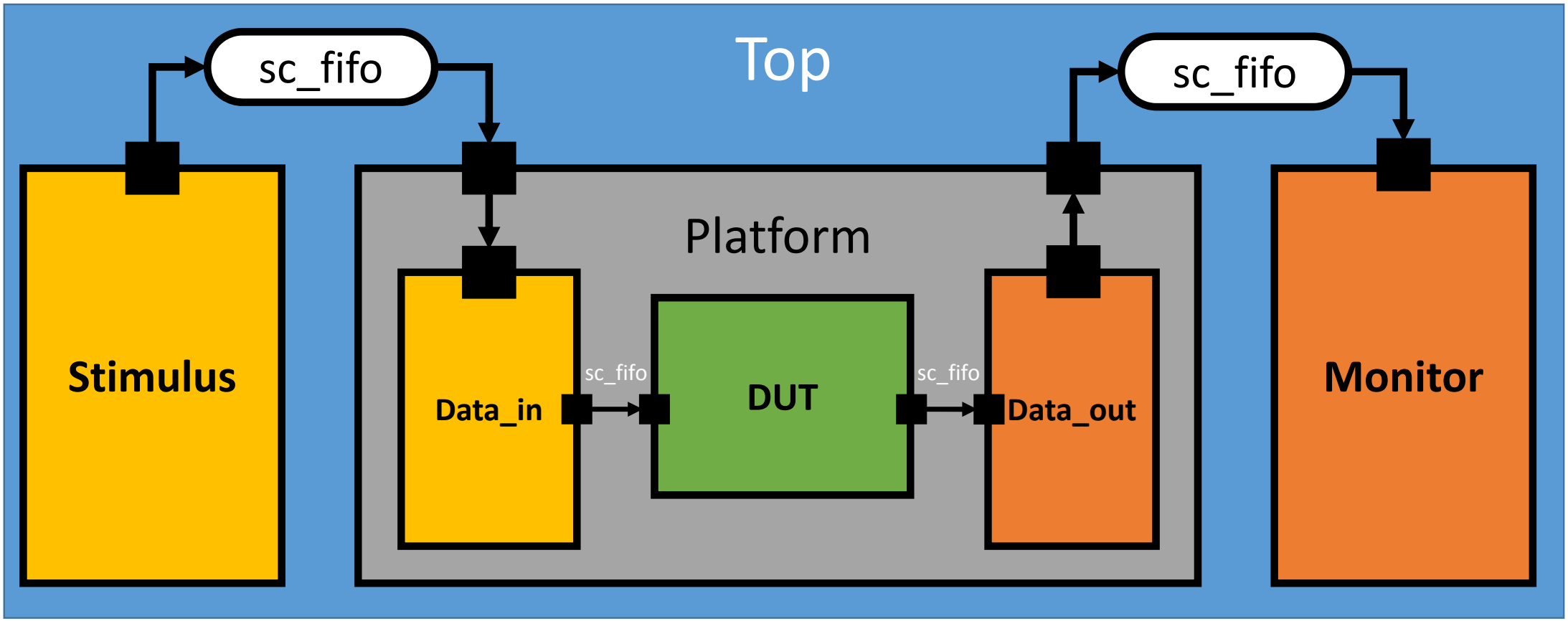
ECPS 203
Discussion

TA: Zhongqi Cheng

Agenda

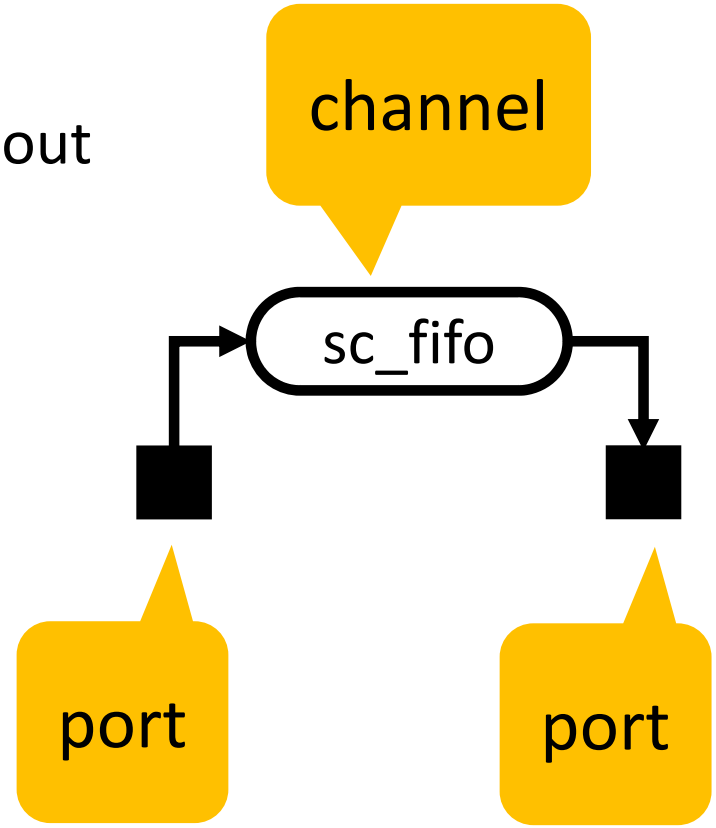
- Build Canny (HW4) in SystemC
  - sc_fifo channel
  - Stimulus
  - Monitor
  - Platform
    - Data_in
    - DUT (design under test)
    - Data_out

Hierarchy


Top
sc_fifo
sc_fifo
Stimulus
Platform
Data_in
sc_fifo
DUT
sc_fifo
Data_out
Monitor

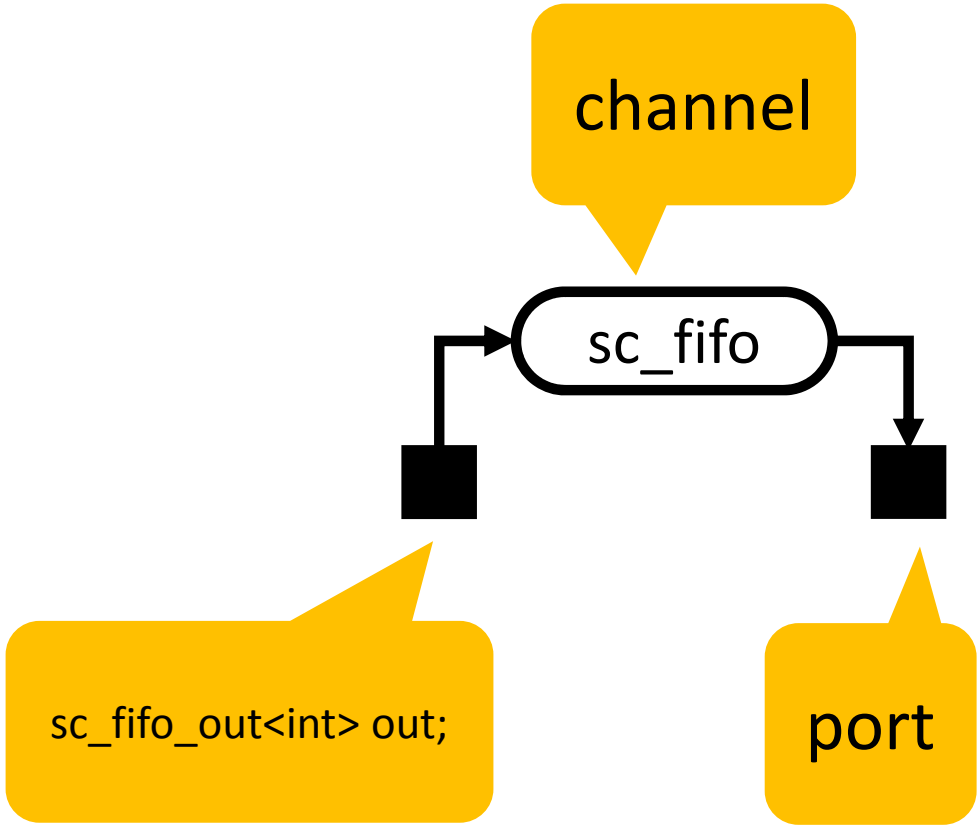
sc_fifo

- FIFO: first in first out



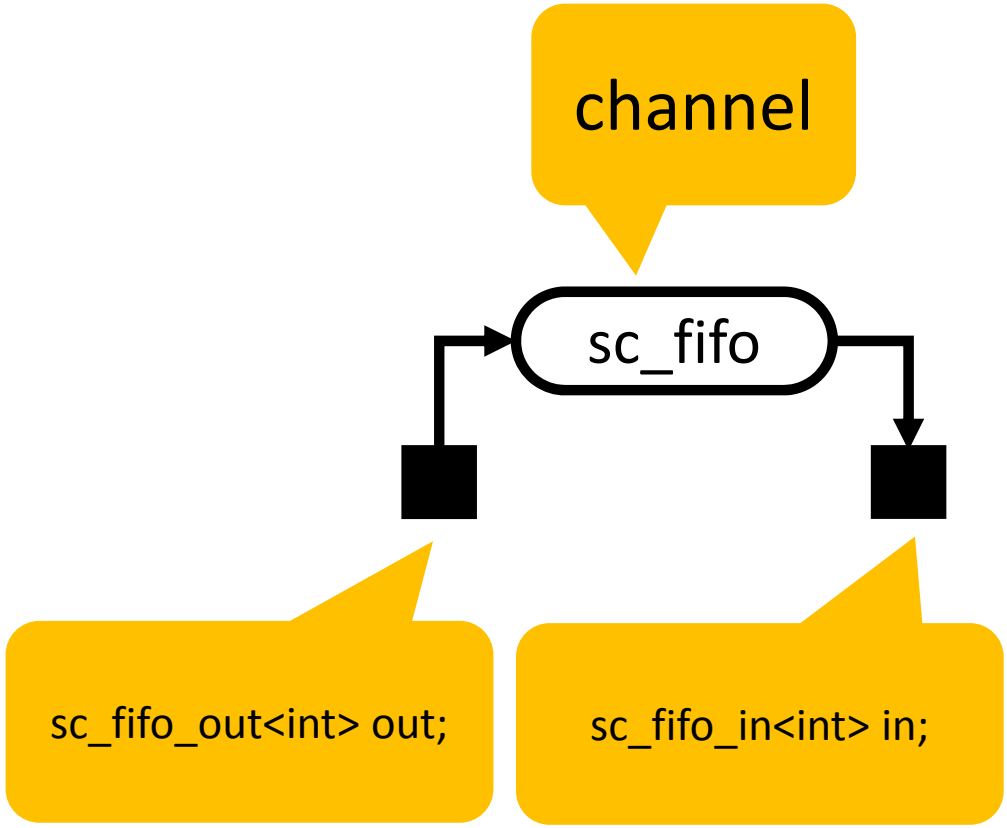
sc_fifo

- output port:
  - sc_fifo_out: output from a module
  - sc_fifo_out<value type> port name;



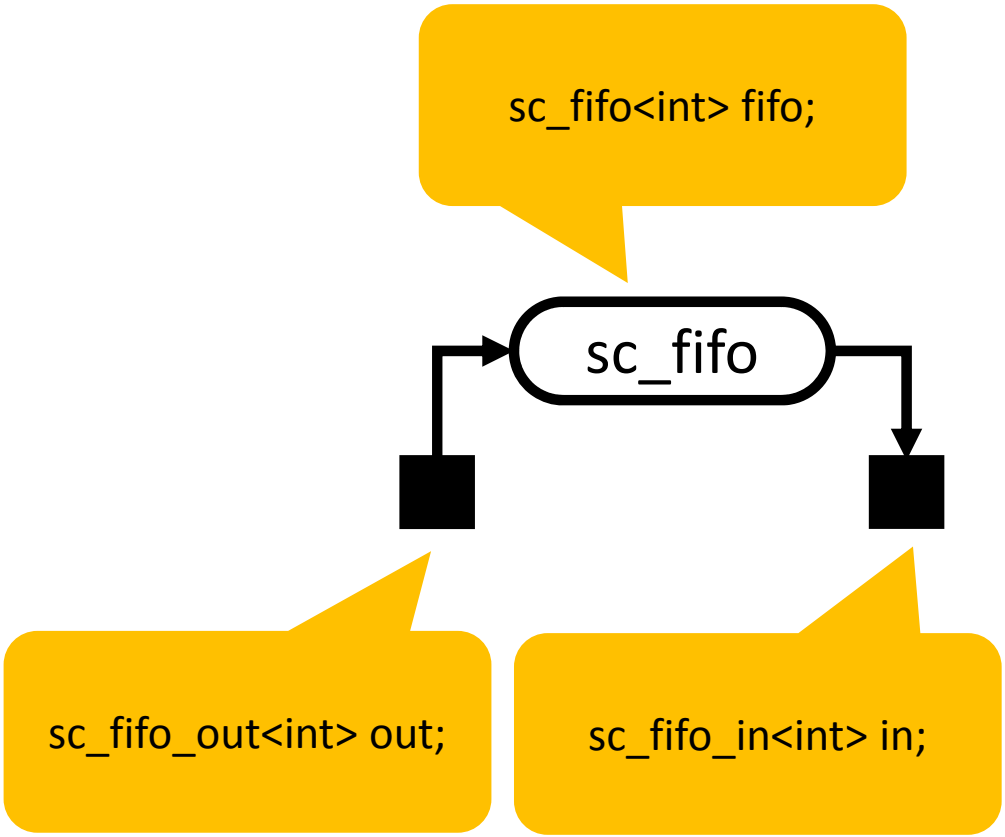
sc_fifo

- output port:
  - sc_fifo_out: output from a module
  - sc_fifo_out<value type> port name;
- input port:
  - sc_fifo_in: input to a module
  - sc_fifo_in<value type> port name;



sc_fifo

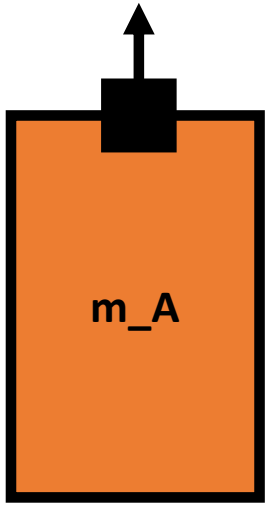
- output port:
  - sc_fifo_out: output from a module
  - sc_fifo_out<value type> port name;
- input port:
  - sc_fifo_in: input to a module
  - sc_fifo_in<value type> port name;
- channel:
  - sc_fifo: channel
  - sc_fifo<value type> channel name;



sc_fifo

- output

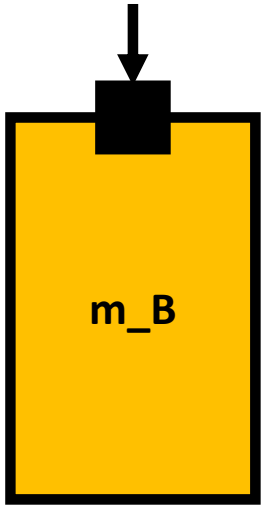
```
int a;
sc_fifo_out<int> out;
out.write(a);
```



sc_fifo

- input

```
int a;
sc_fifo_in<int> in;
in.read(a);
```



sc_fifo

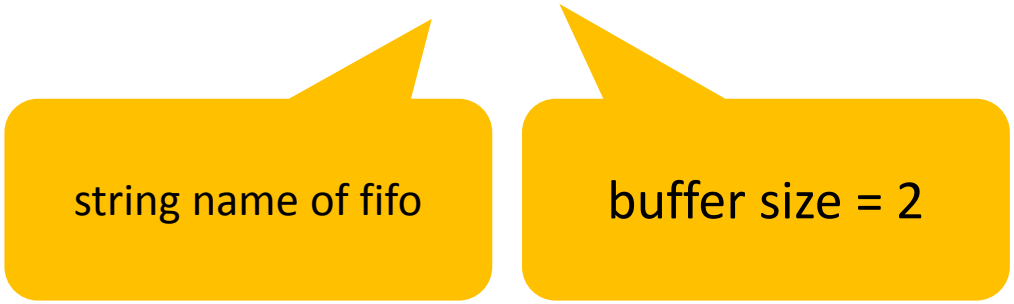
- binding

```
sc_fifo<int> fifo;
m_A.out.bind(fifo);
m_B.in.bind(fifo);
```

sc_fifo

- set the parameters of sc_fifo

```
SC_CTOR(dut)
        : fifo("q1",2)
```



Struct image

- IMAGE: struct image
- is used to wrap the array of the image
- is the data type for communication

- sc_fifo<IMAGE> channel name;
- sc_in<IMAGE> port name;
- sc_out<IMAGE> port name;

```
typedef struct Image_s
{
        unsigned char img[SIZE];

        ...

} IMAGE;
```

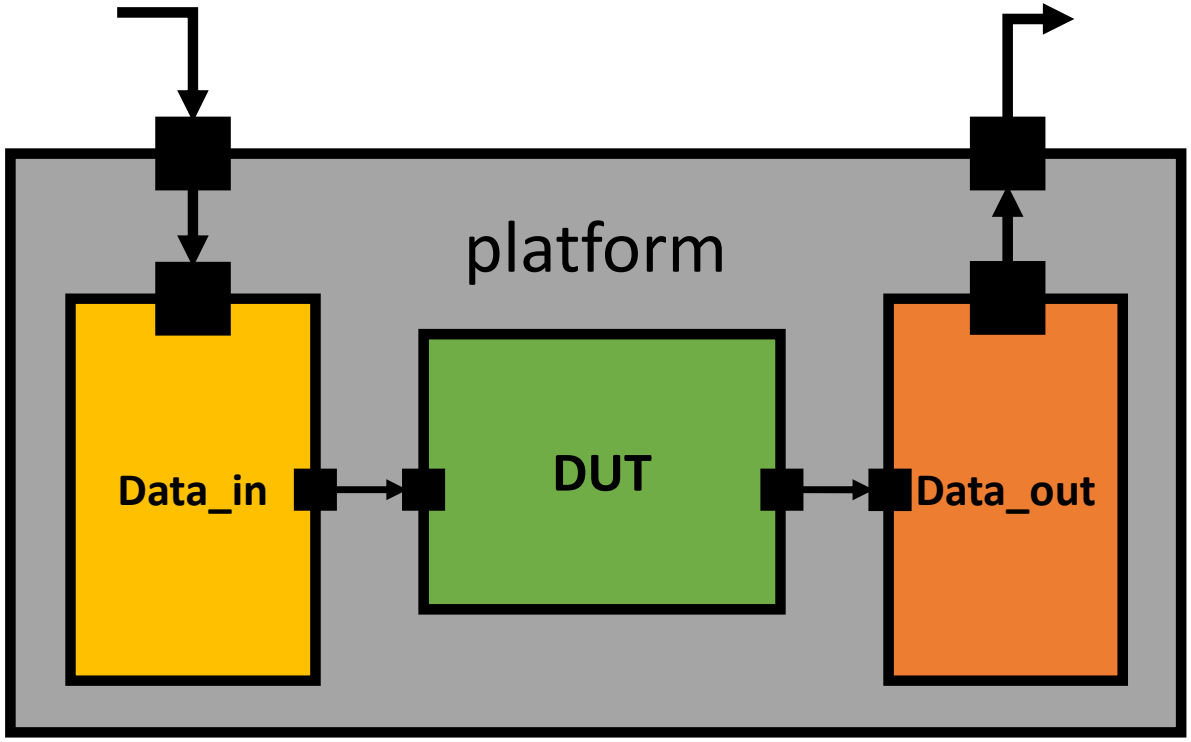
Struct image

- use it as a regular array type

```
IMAGE imageout;
read_pgm_image(infilename, imageout, ROWS, COLS)
out.write(imageout)
```

Platform

- Data_in
- DUT
- Data_out

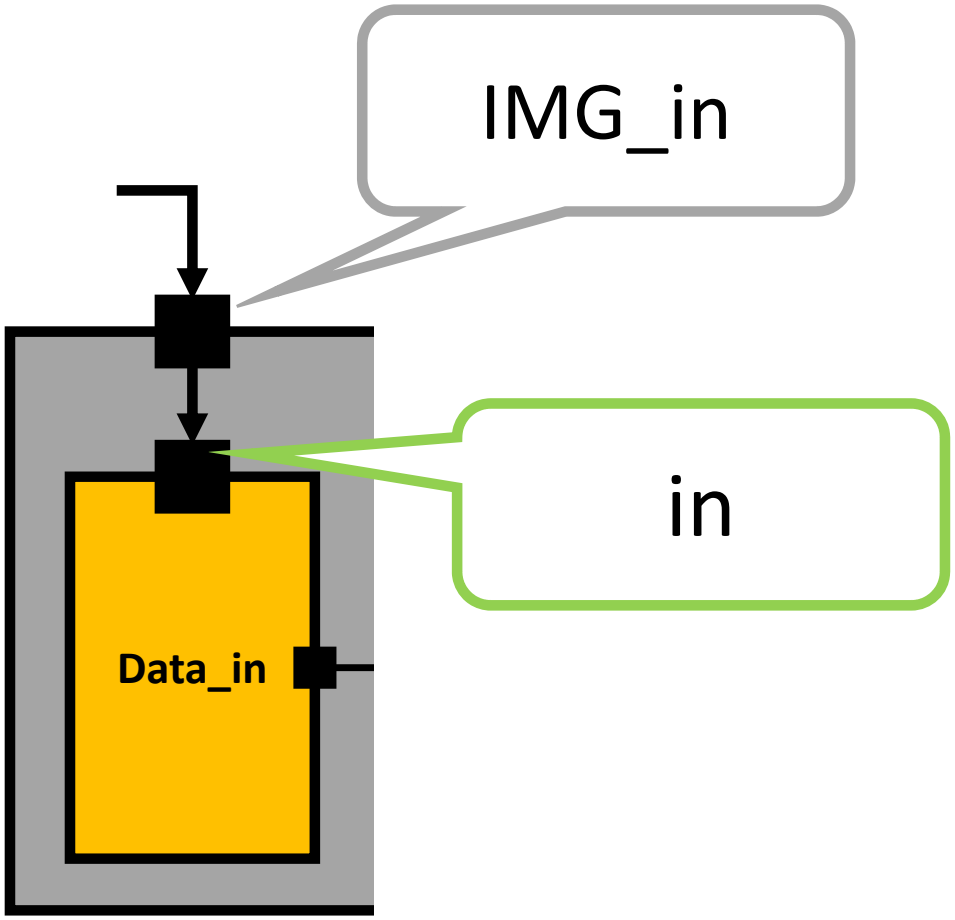


Platform

- Data_in: receive data from outside, send it to DUT
- Data_out: receive data from DUT, send it to outside

Platform

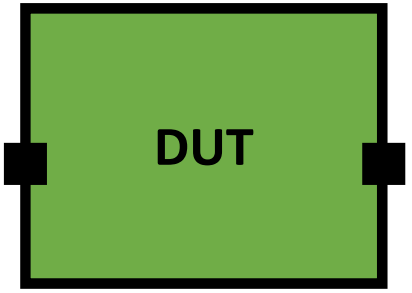
- Data_in.in.bind(IMG_in)



DUT

- performs canny algorithm

1. define a function canny()
2. run all the steps in canny()
3. set canny() as an SC_THREAD



stimulus and monitor

- stimulus:

1. read_pgm_image(…)
2. output the image from port

- monitor:

1. input the image from port
2. write_pgm_image(…)

SC_THREAD

- use SC_THREAD for all modules (except for top and platform)

set stack size

- set_stack_size(128*1024*1024);
- put this code in SC_CTOR

- set stack size in shell as well

Compile

- get Makefile from the public folder
- **make** to compile
- **make test** to run

Submission

- Canny.cpp
- Canny.txt