

# ECPS 203 Discussion Week5

TA: Emad Arasteh

[emalekza@uci.edu](mailto:emalekza@uci.edu)

[ecps203@eecs.uci.edu](mailto:ecps203@eecs.uci.edu)

Office Hours: Fri, 10:00-11:00am

EH 3404 [Zoom 989 2181 4881](https://zoom.us/j/98921814881)

Center for Embedded and Cyber-Physical Systems

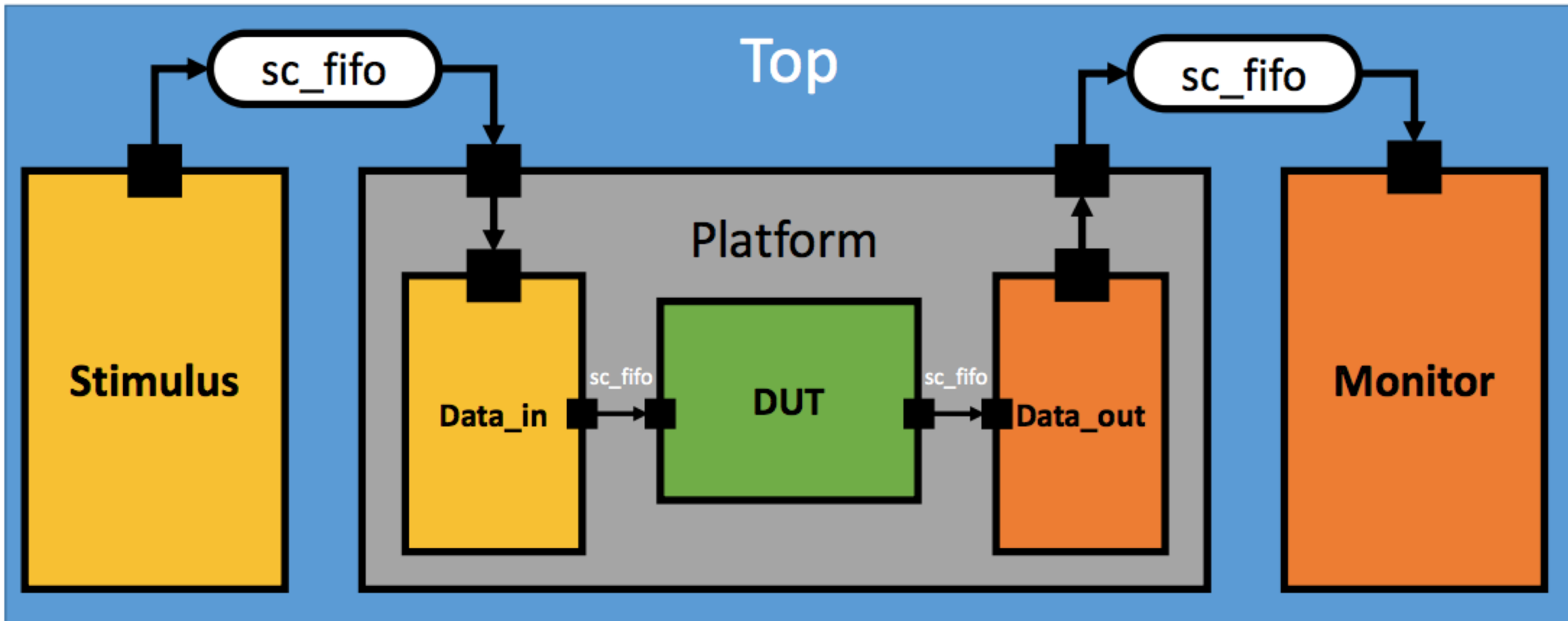
University of California, Irvine



# Outline

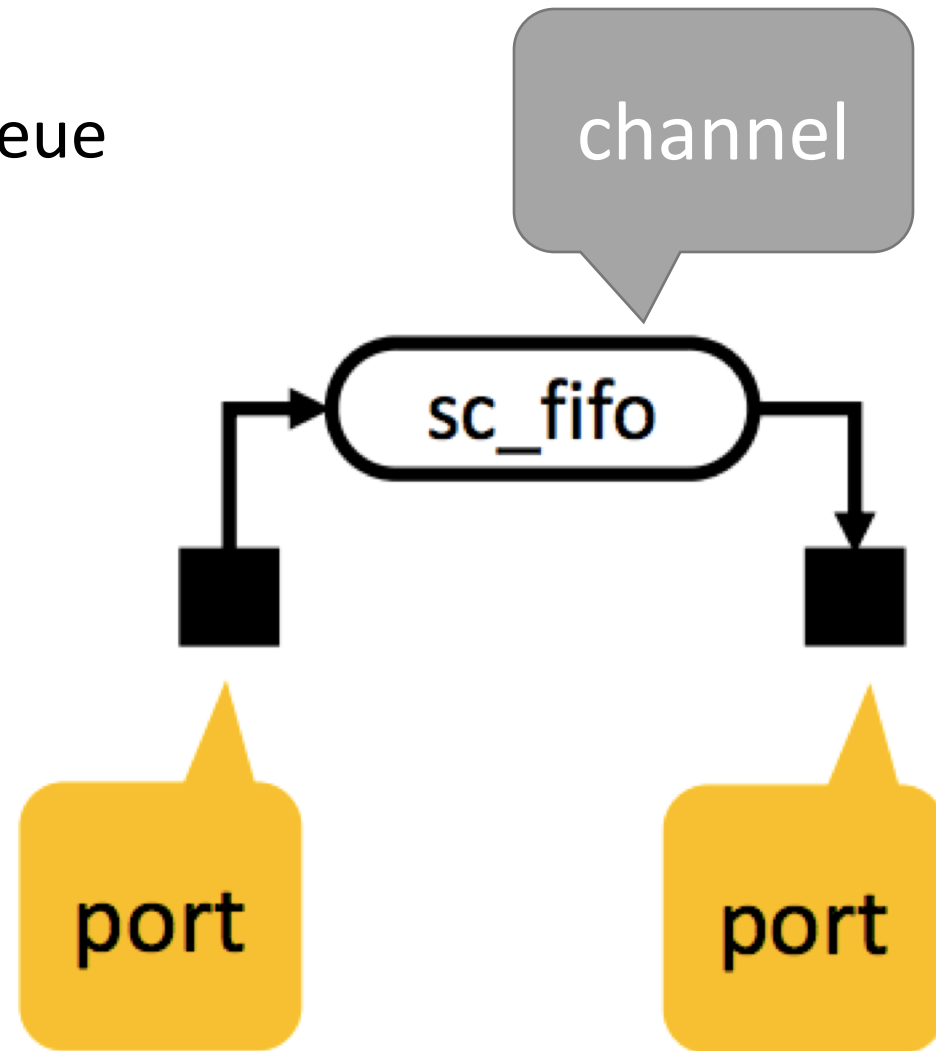
- Assignment 5
  - Test bench model of the Canny Edge Decoder in SystemC
- Test bench structure
- Channel (sc\_fifo)
- Stimulus module
- Monitor module
- Platform
  - Data\_in
  - DUT
  - Data\_out
- Submission
- Questions
- Original slides are prepared by former ECPS 203 TA: Zhongqi Cheng

# Hierarchy



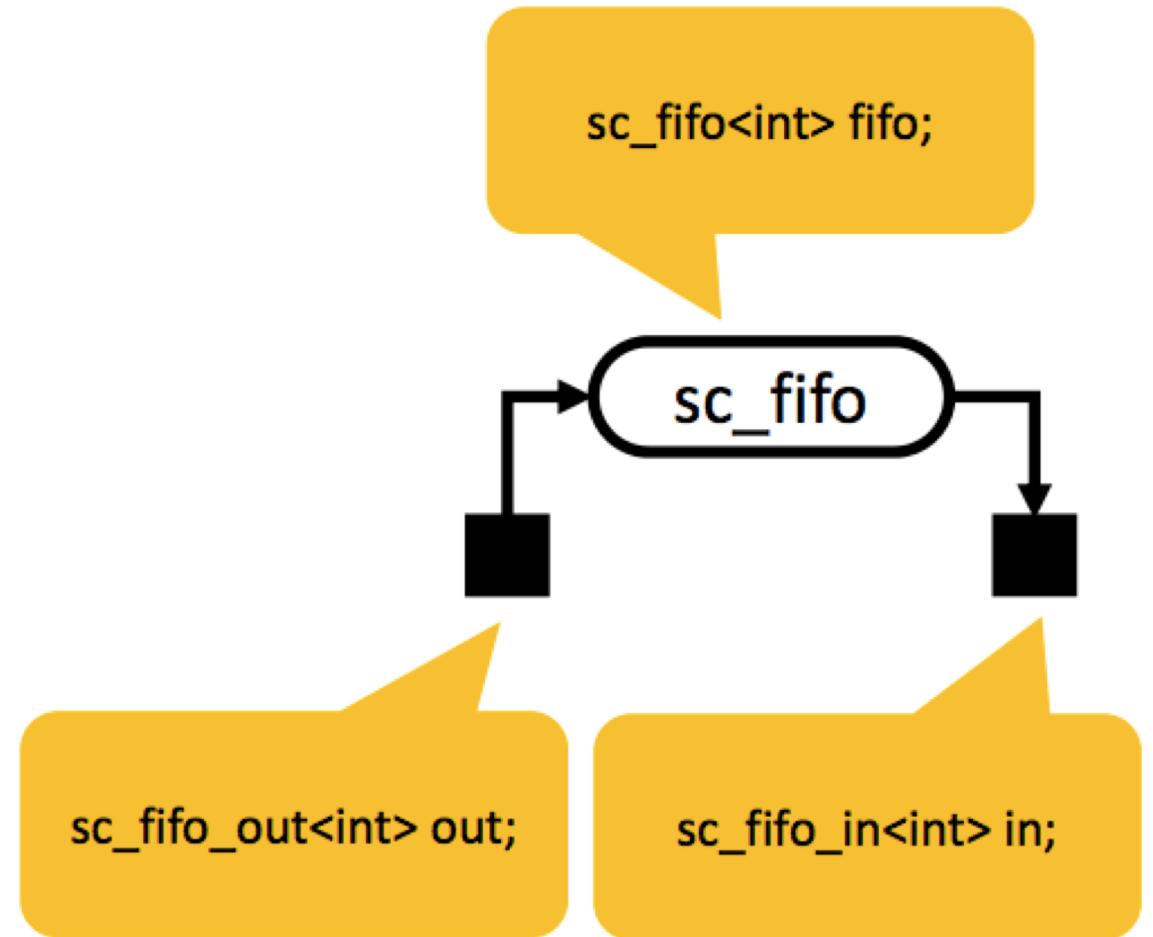
# sc\_fifo

- First in first out queue



# sc\_fifo

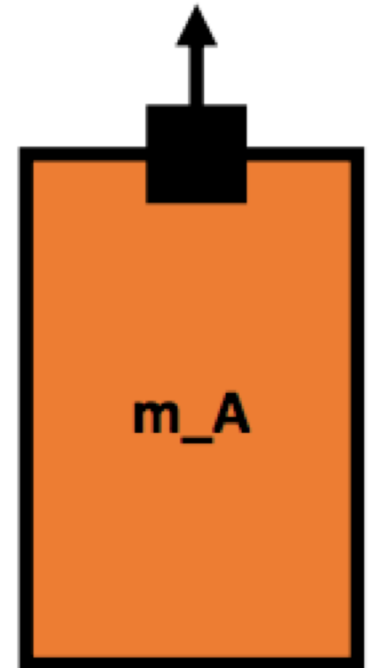
- output port:
  - sc\_fifo\_out: output from a module
  - sc\_fifo\_out ***out***;
- input port:
  - sc\_fifo\_in: input to a module
  - sc\_fifo\_in ***in***;
- channel:
  - sc\_fifo: channel
  - sc\_fifo ***fifo***;



# sc\_fifo

- output from a module: m\_A

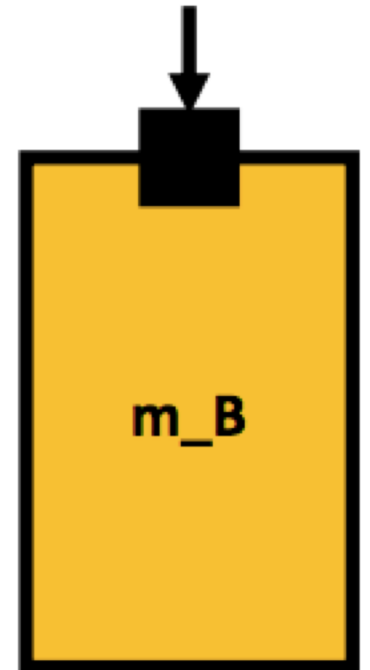
```
int a;  
sc_fifo_out<int> out;  
out.write(a);
```



# sc\_fifo

- input to a module: m\_B

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



# sc\_fifo

- binding in the higher level module

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



# Struct Image\_s

- use as the type parameter of fifo
- wraps an image array
- sc\_fifo fifo;
- sc\_in in;
- sc\_out out;

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

    ...

} IMAGE;
```

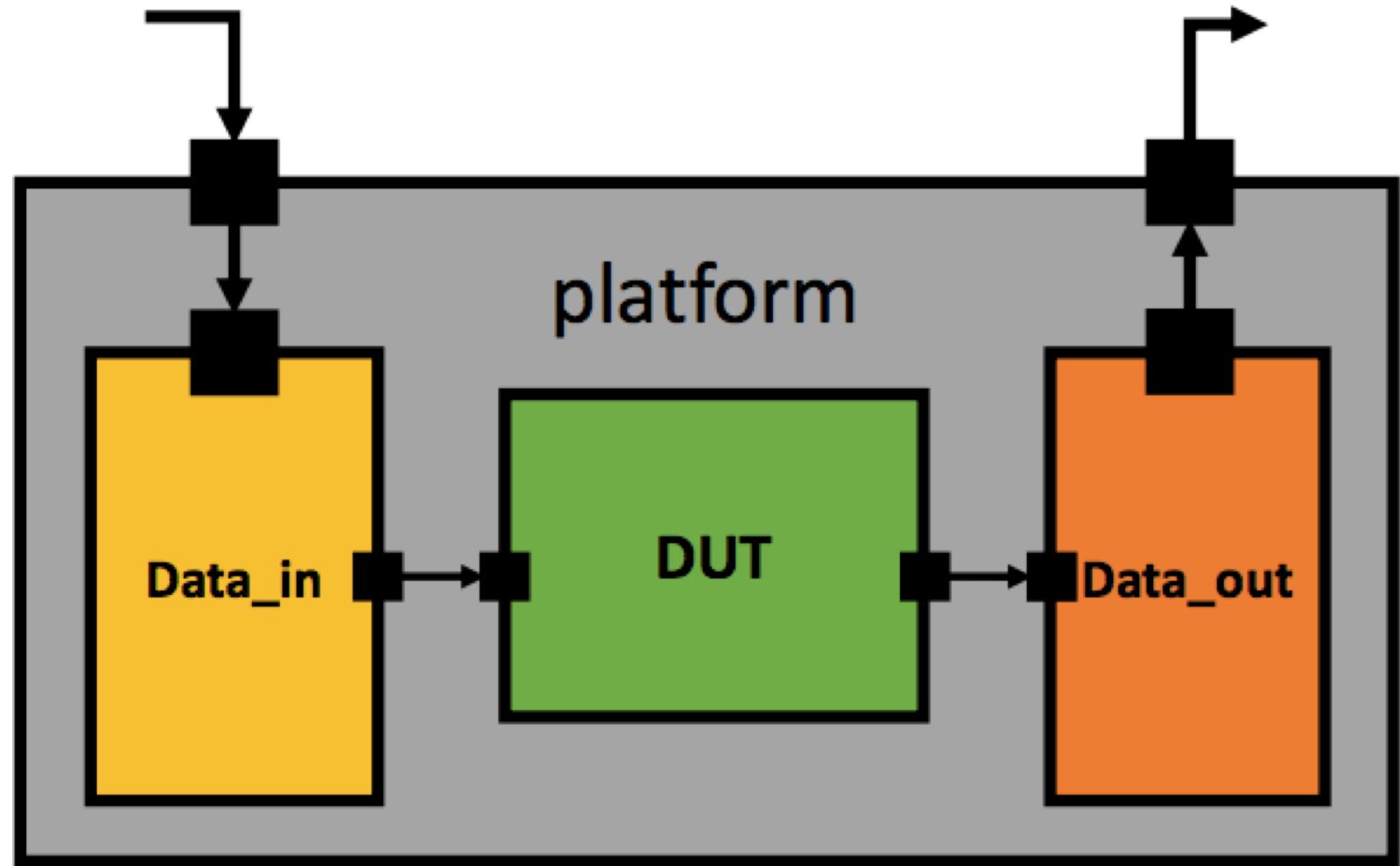
# Struct Image\_s

- use it as a regular array type

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

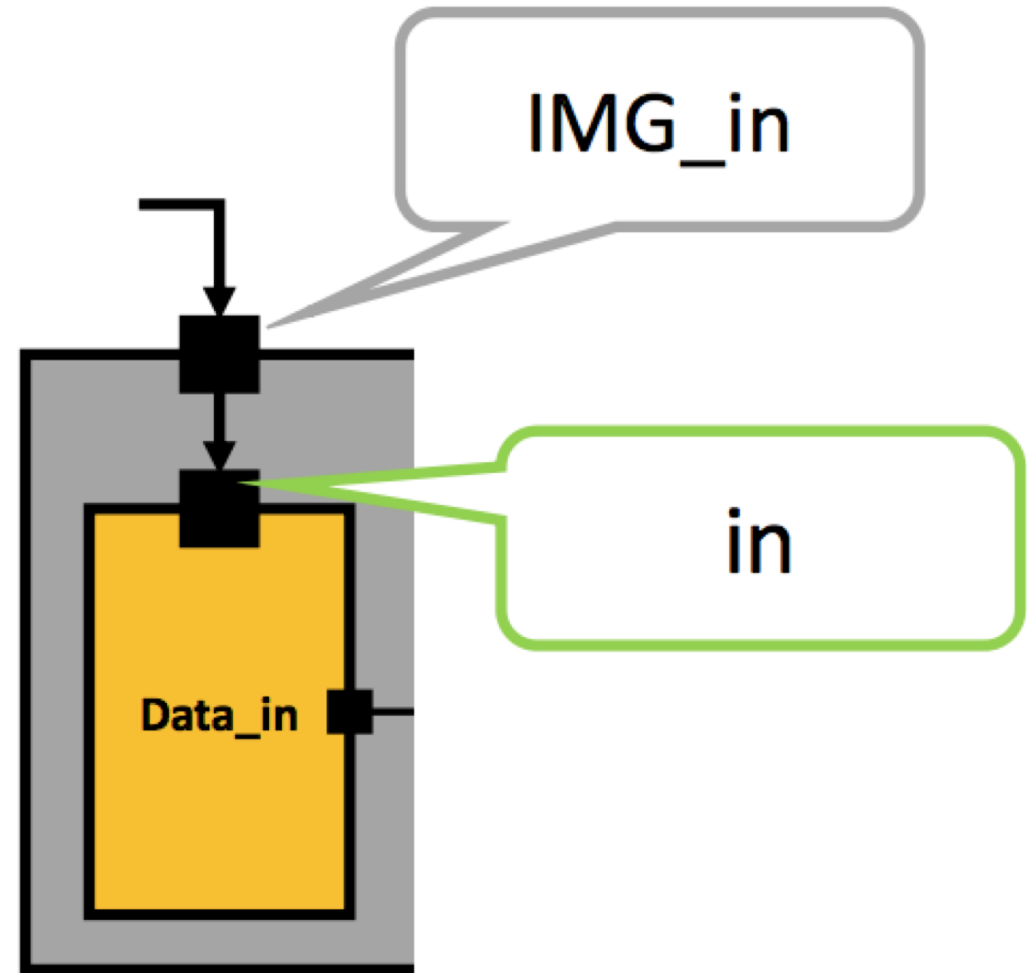
# Platform

- Data\_in
- DUT
- Data\_out



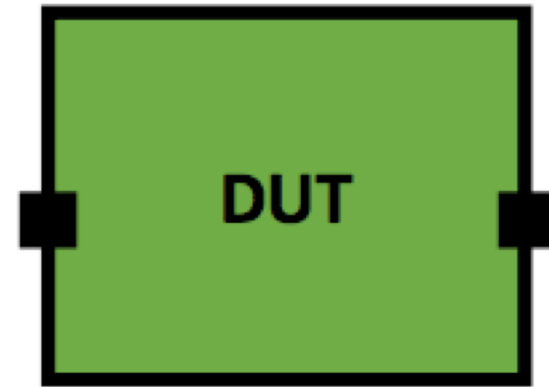
# Platform

- `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(...)`

# set stack size

- `set_stack_size(128*1024*1024);`
- put this code in SC\_CTOR
- set stack size in shell as well

# Compile and run

- `cp ~ecps203/public/MakefileA5 Makefile`
- `make`
- `make test`



# Submission

- Canny.cpp
- Canny.txt