EE6094
CAD for VLSI Design

Checker and Makefile

Andy, Yu-Guang Chen
Assistant Professor, Department of EE
National Central University
andyygchen@ee.ncu.edu.tw
Slides Credit: TA蔡書儀

Outline

- Checker
- Makefile

Outline

- Checker
- Makefile

Checker

- How to use the checker?
  - Step 1: Make sure you put your checker, testcase and your output file testcase.out at the same folder.

```
[ta111521025@eda359_forclass_CAD_PA2]$ ls
checker  testcase1  testcase1.out
```

  - Step 2: Make sure the format of your output file is correct.

Checker

- How to use the checker?
  - Step3: Key in the following commands.

    ./checker testcase testcase.out

```
[ta111521025@eda359_forclass_CAD_PA2]$ ./checker testcase1 testcase1.out
```

  - Step4: If you get the error “permission denied”, key in the following commands.

    chmod 700 checker

```
[109521143@eda359_forclass checkertest]$ ./checker testcase1 testcase1.out
./checker: Permission denied.
[109521143@eda359_forclass checkertest]$ chmod 700 checker
```


Checker

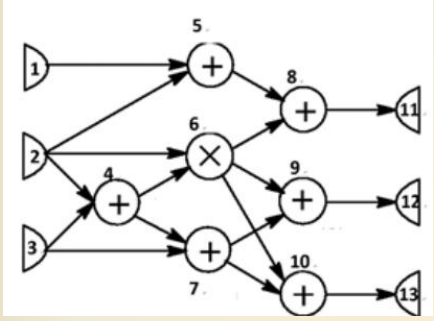
- How to use the checker?
  - Step5: Retry the step 3.

    ./checker testcase testcase.out

```
[109521143@eda359_forclass checkertest]$ ./checker testcase1 testcase1.out
```

Checker

◆Example



Checker

◆Example

➢ If your program is correct, you will get “correct” and “total resource information”.

```
3
1
4  5
6  7
6
6
8  9  10
```

testcase.out

```
[109521143@eda359_forclass checkertest]$ ./checker testcase1 testcase1.out
--------------------CHECKER--------------------
testcase1 is correct!     Resource:  4
```

result

Checker

- Example
  - If your program is wrong, you will get some information "ERROR: the node is not adder or multiplier".

```
3
1
15
4 5
6 7
6
6
8 9 10
```

testcase.out

```
[109521143@eda359_forclass checkertest]$ ./checker testcase1 testcase1.out
--------------------CHECKER--------------------
Error: 15 is not adder or multiplier(time:0)
```

result

Checker

- Example
  - If your program is wrong, you will get some information "Resource-constrain Error".

```
2
1
4 5
6 7
6
6
8 9 10
```

testcase.out

```
[109521143@eda359_forclass checkertest]$ ./checker testcase1 testcase1.out
--------------------CHECKER--------------------
Resourse-constrain Error(time4)
```

result

Checker

- Example
  - If your program is wrong, you will get some information "Adder cycle Error".

```
3
1
4 5
6 7
6 7
6
8 9 10
```

testcase.out

```
[109521143@eda359_forclass checkertest]$ ./checker testcase1 testcase1.out
--------------------CHECKER--------------------
Adder cycle Error: adder7
```

result

Checker

- Example
  - If your program is wrong, you will get some information "Miss Error: the node is not completed".

```
3
1
4 5
6 7
6
6
8 9
```

testcase.out

```
[109521143@eda359_forclass checkertest]$ ./checker testcase1 testcase1.out
--------------------CHECKER--------------------
Miss Error: node10 is not completed
```

result

Checker

- Example
  - If your program is wrong, you will get some information "Dependence Error".

```
3
1
4 7
6 5
6
6
8 9 10
```

testcase.out

```
[109521143@eda359_forclass checkertest]$ ./checker testcase1 testcase1.out
--------------------CHECKER--------------------
Dependence Error(4and7)
```

result

Checker

- Example
  - If your program is wrong, you will get some information "Latency Error".

latency constrain: 5

```
3
1
cycle 1  4 5
cycle 2  6 7
cycle 3  6
cycle 4  6
cycle 5  8 9
cycle 6  10
```

testcase.out

```
[109521143@eda359_forclass checkertest]$ ./checker testcase1 testcase1.out
--------------------CHECKER--------------------
Latency Error
```

result

Outline

- Checker
- Makefile

Makefile

- In demo session, TA will test your makefile in the following commands.
  - *make all*
  - *make run Testcase=testcase*
  - *make clean*

Makefile

- *make all*

```
[ta111521025@eda359_forclass List_code]$ ls
checker   PA2_circuit.cpp  PA2_main.cpp  PA2_node.h
Makefile  PA2_circuit.h    PA2_node.cpp  testcase1
[ta111521025@eda359_forclass List_code]$ make all
[ta111521025@eda359_forclass List_code]$ ls
checker   PA2_circuit.cpp  PA2_main.cpp  PA2_node.h
Makefile  PA2_circuit.h    PA2_main.o    PA2_node.o
PA2       PA2_circuit.o    PA2_node.cpp  testcase1
[ta111521025@eda359_forclass List_code]$
```

After *make all* , an executable file is generated

Makefile

- *make run Testcase=testcase1*

```
[ta111521025@eda359_forclass List_code]$ ls
checker   PA2_circuit.cpp  PA2_main.cpp  PA2_node.h  testcase2
Makefile  PA2_circuit.h    PA2_main.o    PA2_node.o  testcase3
PA2       PA2_circuit.o    PA2_node.cpp  testcase1   testcase4
[ta111521025@eda359_forclass List_code]$ make run Testcase=testcase1
[ta111521025@eda359_forclass List_code]$ ls
checker   PA2_circuit.cpp  PA2_main.cpp  PA2_node.h  testcase1.out
Makefile  PA2_circuit.h    PA2_main.o    PA2_node.o  testcase2
PA2       PA2_circuit.o    PA2_node.cpp  testcase1   testcase3
[ta111521025@eda359_forclass List_code]$
```

Scheduling result of testcase1 after *make run*

Makefile

- *make clean*

```
[ta111521025@eda359_forclass List_code]$ ls
checker   PA2_circuit.cpp  PA2_main.cpp  PA2_node.h
Makefile  PA2_circuit.h    PA2_main.o    PA2_node.o
PA2       PA2_circuit.o    PA2_node.cpp  testcase1
[ta111521025@eda359_forclass List_code]$ make clean
[ta111521025@eda359_forclass List_code]$ ls
checker   PA2_circuit.cpp  PA2_main.cpp  PA2_node.h
Makefile  PA2_circuit.h    PA2_node.cpp  testcase1
[ta111521025@eda359_forclass List_code]$
```