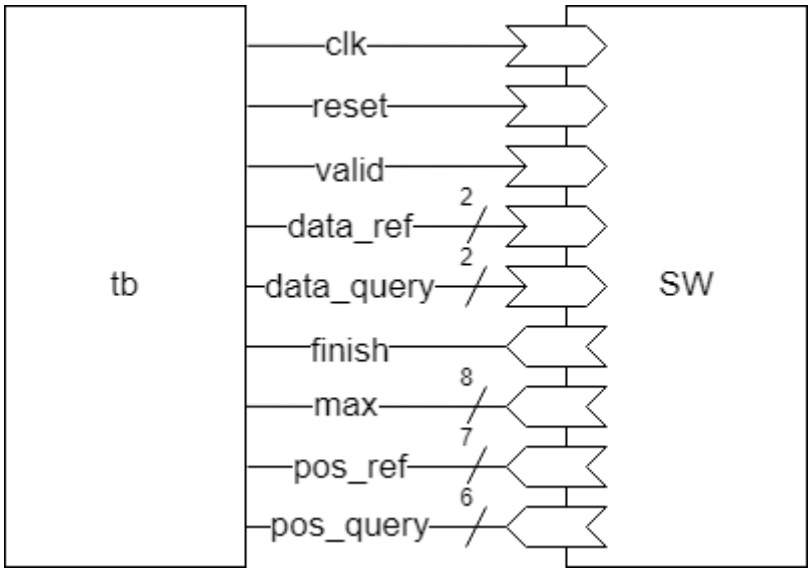
ICD Project --- Smith-Waterman Algorithm with Multiple PEs

Due: 2022/06/17(Fri) 14:00

1. Introduction

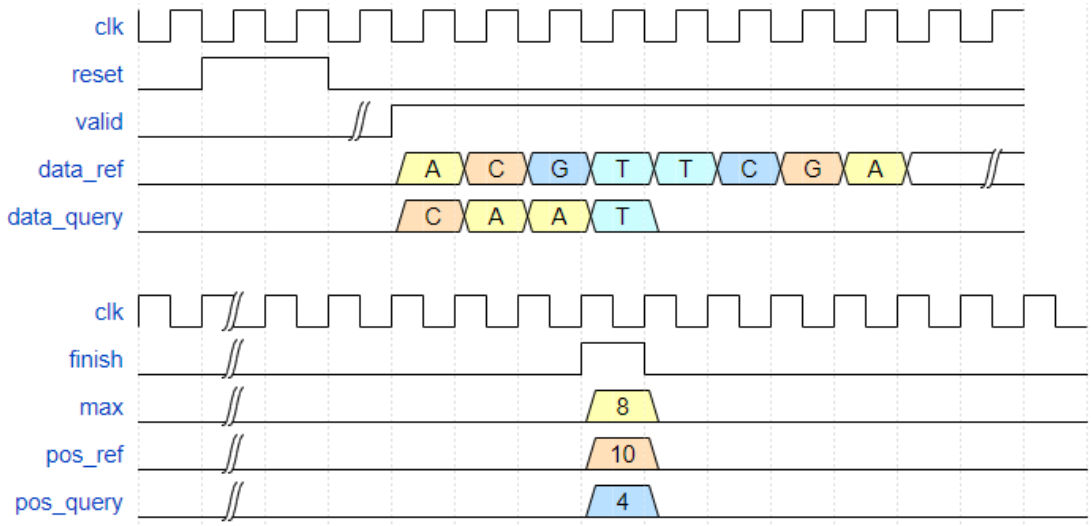
In this project, you are asked to finish the same algorithm as HW4 & HW5, but more complicated. The reference length and query length are larger, and you need to consider the performance of your design this time. The reference length is scaled up to 64, and the query length is scaled up to 48. You can use at most 16 PEs (processing elements, further introduction in *3. Function Description*) in your design, which will improve your time cost.

2. Specification



Name	I/O	Width	Description
clk	*input*	1	Clock signal. This design is positive-edge-triggered.
reset	*input*	1	Active high asynchronous system reset signal.
valid	*input*	1	If it is set to high, the transferred *data_ref* and *data_query* are valid
data_ref	*input*	2	The data of reference sequence will be sent when *valid* is set to high. (0(A) 1(C) 2(G) 3(T))
data_query	*input*	2	The data of query sequence will be sent when *valid* is set to high. (0(A) 1(C) 2(G) 3(T))
finish	*output*	1	When *finish* is set to high, the testbench will examine the signal of *max*, *pos_ref* and *pos_query*
max	*output*	8	The highest score of SW algorithm. (H table)
pos_ref	*output*	7	The corresponding position on reference sequence of the highest score.
pos_query	*output*	6	The corresponding position on query sequence of the highest score.

3. Function Description



For the charts below, we show some selected time stamps for a multiple-PE design. White cells represent that the cells have not been processed yet, while green cells indicate that the cells have been processed. The blue cell is the active cell processed by the processing element (PE) at time T. Note that you can use at most 16 PEs in your design. (Showing with ref_len = 16 / query_len = 4 / 2 PEs as an example)

T=1

	i=0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
j=0	Q\R	A	C	G	T	T	G	C	A	A	T	C	G	T	A	G	C
1	C																
2	A																
3	A																
4	T																

T=2

	i=0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
j=0	Q\R	A	C	G	T	T	G	C	A	A	T	C	G	T	A	G	C
1	C																
2	A																
3	A																
4	T																

T=16

	i=0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
j=0	Q\R	A	C	G	T	T	G	C	A	A	T	C	G	T	A	G	C
1	C																
2	A																
3	A																
4	T																

T=17

	i=0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
j=0	Q\R	A	C	G	T	T	G	C	A	A	T	C	G	T	A	G	C
1	C																
2	A																
3	A																
4	T																

T=18

	i=0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
j=0	Q\R	A	C	G	T	T	G	C	A	A	T	C	G	T	A	G	C
1	C																
2	A																
3	A																
4	T																

T=32

	i=0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
j=0	Q\R	A	C	G	T	T	G	C	A	A	T	C	G	T	A	G	C
1	C																
2	A																
3	A																
4	T																

T=33

	i=0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
j=0	Q\R	A	C	G	T	T	G	C	A	A	T	C	G	T	A	G	C
1	C																
2	A																
3	A																
4	T																

4. Files

File Name	Description
tb.v	Testbench to test your design
SW.v	Design under test, don't change I/O interface of the module
./dat/ref.dat	Test data as reference
./dat/query.dat	Test data as query
sythesis.tcl	Design constraint file for synthesis. You can change the cycle time in this file to fit your own design.
.synopsys_dc.setup	Environment setting file for design compiler (DON'T change)
tsmc13.v	Process file for synthesis
./layout/SW_APR.sdc	Design constraint file for APR. You can change the cycle time in this file to fit your own design.
./layout/ & ./library/	Files for APR (DON'T change)
./syn/tsmc13_neg.v	Process file for APR

5. RTL Simulation Scripts

One testbench is provided. The simulation command is as below:

```
ncverilog tb.v SW.v +define+tb1
```

If you want to export the waveform, you can use `+define+FSDB` or `+define+VCD` command with `+access+r`.

```
ncverilog tb.v SW.v +define+tb1+FSDB +access+r
ncverilog tb.v SW.v +define+tb1+VCD +access+r
```

6. Synthesis Simulation Scripts

After synthesis, the simulation command is as below:

- `ncverilog tb.v SW_syn.v tsmc13.v +define+tb1+SDFSYN`

If you want to export the waveform, you can use `+define+FSDB` or `+define+VCD` command with `+access+r`.

- `ncverilog tb.v SW_syn.v tsmc13.v +define+tb1+FSDB+SDFSYN +access+r`
- `ncverilog tb.v SW_syn.v tsmc13.v +define+tb1+VCD+SDFSYN +access+r`

Noted that you can change the cycle time in testbench by yourself. The cycle time used in testbench must be equal or bigger than the value used in .tcl file during synthesis.

7. APR Simulation Scripts

After APR, the simulation command is as below:

- `ncverilog tb.v SW_APR.v -v ./syn/tsmc13_neg.v +define+tb1+SDFAPR +ncmaxdelays`

If you want to export the waveform, you can use `+define+FSDB` or `+define+VCD` command with `+access+r`.

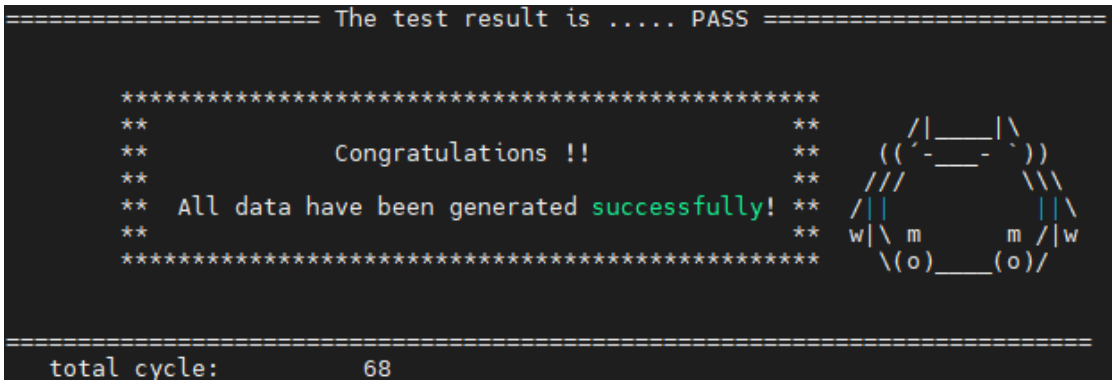
- `ncverilog tb.v SW_APR.v -v ./syn/tsmc13_neg.v +define+tb1+FSDB+SDFAPR +ncmaxdelays +access+r`
- `ncverilog tb.v SW_APR.v -v ./syn/tsmc13_neg.v +define+tb1+VCD+SDFAPR +ncmaxdelays +access+r`

Noted that you can change the cycle time in testbench by yourself. The cycle time used in testbench must be equal or bigger than the value used in .sdc file during APR.

If there are some errors about `ANTENNA` during simulation, you can comment the line with `ANTENNA` in `SW_APR.v` to pass the simulation.

8. Simulation Results

If all the results are correct, you will see a Snorlax.



9. Homework Requirements

a. Pass 2 testbench of RTL simulation (30%)

b. Pass 2 testbench of gate level simulation (with no latches) (20%)

c. Pass 2 testbench + 1 hidden testbench of APR simulation (10%)

(1) With no error after verify geometry

(2) Power ring: width = 2, ring number $\geq$ 2

(3) Power stripe: width = 1, stripe number $\geq$ 1

d. Report (10%)

If you don't pass requirement c, the score will be determined by completeness of your report; else, you can only write a simple report as you've done in HW5 to get the full report score.

e. APR performance passing testbench 1 (30%)

Cost = Area * Timing^2 (um^2 * ns^2)

(A design with a lower cost is considered a better design.)

Area = Total area of Core in summaryReport.rpt

Timing = Runtime of testbench 1 (for example: 315 ns)

```
Simulation complete via $finish(1) at time 315 NS + 0
```

The final score is according to your ranking:

1st place to 6th place: 30, 28, 26, 24, 22, 20 points

7th place to the last place: 19, 18, 17, 16, … (and so on)

f. Compress the files to `FINAL_bXX901XXX_bXX901XXX.zip` and submit it.

```
FINAL_bXX901XXX_bXX901XXX.zip/
FINAL_bXX901XXX_bXX901XXX/
      - SW.v
      - SW_syn.v
      - SW_syn.sdf
      - SW_syn.ddc
      - SW_timing.txt
      - SW_area.txt
      - SW_power.txt
      - SW_APR.v
      - SW_APR.sdf
      - summaryReport.rpt
      - bXX901XXX_bXX901XXX_report.pdf
```

10.Deadline

Please submit the .zip file to Ceiba before 2022/06/17 (Fri) 14:00.

If there is any problem, feel free to contact TAs by emails. Do not forget to add [積體電路設計] in your email title.

For Verilog problems:

林奕憲 d06943006@ntu.edu.tw

葉陽明 d05943006@ntu.edu.tw

For algorithm problems:

黃士豪 r10943004@ntu.edu.tw

沈哲瑋 r10943013@ntu.edu.tw