2024 Digital IC Design
Homework 4: Max-Priority Queue

NAME	楊晴雯
Student ID	P76114511

Simulation Result

Functional simulation	**100**	Gate-level simulation	Score	Clock width	15.0 (ns)	Gate-level simulation time	See below.

P0

```
VSIM 509> run -all
# cycle=      4,data=00
# cycle=      5,data=02
# cycle=      6,data=05
# cycle=      7,data=07
# cycle=      8,data=09
# cycle=      9,data=64
# cycle=     10,data=32
# cycle=     11,data=fa
# cycle=     12,data=15
# cycle=     13,data=36
# cycle=     14,data=06
# cycle=     15,data=4f
# output 4f,expect 4f
# output 36,expect 36
# output 32,expect 32
# output 15,expect 15
# output 06,expect 06
# output 05,expect 05
# output 07,expect 07
# output 02,expect 02
# output 09,expect 09
#  ****************************
#  **                        **
#  **  Congratulations !!    **
#  **                        **       |__||
#  **  Simulation PASS !!    **      / O.O  |
#  **                        **    /_____   |
#  **  Your score =100       **   /^ ^ ^ \  |
#  **                        **  |^ ^ ^ ^ |w|
#  ****************************   \m___m__|_|
# ** Note: $finish    : testfixture.v(180)
#    Time: 1630 ns  Iteration: 0  Instance: /test
```

P1

```
# output fa,expect fa
# output 64,expect 64
# output 50,expect 50
# output 36,expect 36
# output 4f,expect 4f
# output 32,expect 32
# output 07,expect 07
# output 15,expect 15
# output 09,expect 09
# output 06,expect 06
# output 05,expect 05
# output 02,expect 02
#  ****************************
#  **                        **
#  **  Congratulations !!    **
#  **                        **       |__||
#  **  Simulation PASS !!    **      / O.O  |
#  **                        **    /_____   |
#  **  Your score =100       **   /^ ^ ^ \  |
#  **                        **  |^ ^ ^ ^ |w|
#  ****************************   \m___m__|_|
# ** Note: $finish    : testfixture.v(180)
```

P2

P0 (2416.806 ns)

```
# output 4f,expect 4f
# output 36,expect 36
# output 32,expect 32
# output 15,expect 15
# output 06,expect 06
# output 05,expect 05
# output 07,expect 07
# output 02,expect 02
# output 09,expect 09
#  ****************************
#  **                        **
#  **  Congratulations !!    **
#  **                        **       |__||
#  **  Simulation PASS !!    **      / O.O  |
#  **                        **    /_____   |
#  **  Your score =100       **   /^ ^ ^ \  |
#  **                        **  |^ ^ ^ ^ |w|
#  ****************************   \m___m__|_|
# ** Note: $finish    : testfixture.v(180)
#    Time: 2416806 ps  Iteration: 0  Instance: /test
```

P1 (2731.806 ns)

```
# output 50,expect 50
# output 37,expect 37
# output 34,expect 34
# output 36,expect 36
# output 06,expect 06
# output 05,expect 05
# output 32,expect 32
# output 15,expect 15
# output 02,expect 02
#  ****************************
#  **                        **
#  **  Congratulations !!    **
#  **                        **       |__||
#  **  Simulation PASS !!    **      / O.O  |
#  **                        **    /_____   |
#  **  Your score =100       **   /^ ^ ^ \  |
#  **                        **  |^ ^ ^ ^ |w|
#  ****************************   \m___m__|_|
# ** Note: $finish    : testfixture.v(180)
#    Time: 2731806 ps  Iteration: 0  Instance: /test
```

P2 (3301.806 ns)

```
# cycle=     14,data=06
# cycle=     15,data=4f
# output 4f,expect 4f
# output 36,expect 36
# output 32,expect 32
# output 15,expect 15
# output 06,expect 06
# output 05,expect 05
# output 07,expect 07
# output 02,expect 02
# output 09,expect 09
#  ****************************
#  **                        **
#  **  Congratulations !!    **
#  **                        **       |__||
#  **  Simulation PASS !!    **      / O.O  |
#  **                        **    /_____   |
#  **  Your score =100       **   /^ ^ ^ \  |
#  **                        **  |^ ^ ^ ^ |w|
#  ****************************   \m___m__|_|
# ** Note: $finish    : testfixture.v(180)
#    Time: 1630 ns  Iteration: 0  Instance: /test
```

```
# output 64,expect 64
# output 37,expect 37
# output 50,expect 50
# output 36,expect 36
# output 06,expect 06
# output 45,expect 45
# output 34,expect 34
# output 15,expect 15
# output 1b,expect 1b
# output 02,expect 02
# output 05,expect 05
# output 32,expect 32
# output 44,expect 44
#  ****************************
#  **                        **
#  **  Congratulations !!    **
#  **                        **       |__||
#  **  Simulation PASS !!    **      / O.O  |
#  **                        **    /_____   |
#  **  Your score =100       **   /^ ^ ^ \  |
#  **                        **  |^ ^ ^ ^ |w|
#  ****************************   \m___m__|_|
# ** Note: $finish    : testfixture.v(180)
#    Time: 3301806 ps  Iteration: 0  Instance: /test
```

P3

```
# cycle=     14,data=06
# cycle=     15,data=4f
# output 50,expect 50
# output 37,expect 37
# output 34,expect 34
# output 36,expect 36
# output 06,expect 06
# output 05,expect 05
# output 32,expect 32
# output 15,expect 15
# output 02,expect 02
#  ****************************
#  **                        **
#  **  Congratulations !!    **
#  **                        **       |__||
#  **  Simulation PASS !!    **      / O.O  |
#  **                        **    /_____   |
#  **  Your score =100       **   /^ ^ ^ \  |
#  **                        **  |^ ^ ^ ^ |w|
#  ****************************   \m___m__|_|
# ** Note: $finish    : testfixture.v(180)
#    Time: 1810 ns  Iteration: 0  Instance: /test
```

P3 (3301.806 ns)

```
# output 64,expect 64
# output 37,expect 37
# output 50,expect 50
# output 36,expect 36
# output 06,expect 06
# output 45,expect 45
# output 34,expect 34
# output 15,expect 15
# output 1b,expect 1b
# output 02,expect 02
# output 05,expect 05
# output 32,expect 32
# output 44,expect 44
#  ****************************
#  **                        **
#  **  Congratulations !!    **
#  **                        **       |__||
#  **  Simulation PASS !!    **      / O.O  |
#  **                        **    /_____   |
#  **  Your score =100       **   /^ ^ ^ \  |
#  **                        **  |^ ^ ^ ^ |w|
#  ****************************   \m___m__|_|
# ** Note: $finish    : testfixture.v(180)
#    Time: 3301806 ps  Iteration: 0  Instance: /test
```

Synthesis Result

Total logic elements	3,365 / 55,856 (6%)
Total memory bit	0 / 2,396,160 (0%)
Embedded multiplier 9-bit element	0 / 308 (0%)

Flow Status	Successful - Sat May 25 17:07:17 2024
Quartus Prime Version	20.1.1 Build 720 11/11/2020 SJ Lite Edition
Revision Name	MPQ
Top-level Entity Name	MPQ
Family	Cyclone IV E
Device	EP4CE55F23A7
Timing Models	Final
Total logic elements	3,365 / 55,856 (6 %)
Total registers	435
Total pins	50 / 325 (15 %)
Total virtual pins	0
Total memory bits	0 / 2,396,160 (0 %)
Embedded Multiplier 9-bit elements	0 / 308 (0 %)
Total PLLs	0 / 4 (0 %)

Description of your design

Some of the principles in mind in planning the design are:
1. No while or for loops (certainly no recursion). Use state transition to replace.
2. In one state, do one job: e.g. a constant-time comparison or value exchange.
For example, the core function “MAX_HEAPIFY” is implemented using 3 states: _FIND_LARGEST, _EXCHANGE, DONE_MAX_HEAPIFY. In the first state I compare the index’s corresponding value and its left child and right child to get the largest-value index; in _EXCHANGE I change the value, DONE_MAX_HEAPIFY signals the process of MAX_HEAPIFY function is done (resembling the “return to the caller” concept in software; like the recursion call stack is emptied).
I ended up designing 14 states for this data structure. See the following state-transition diagram. My design has a clock cycle-width of 15 ns and passes all of the testcases within 1000 cycles.

*Scoring = (Total logic elements + total memory bit + 9*embedded multiplier 9-bit element)* × *(Total cycle used*clock width)*