# 2024 Digital IC Design
# Homework 4: Max-Priority Queue

| NAME | 陳穎祺 |
|---|---|
| Student ID | P76121615 |

## Simulation Result

| Functional simulation | 100 | Gate-level simulation | 100 | Clock width | 25ns | Gate-level simulation time | 4430.061**ns** |
|---|---|---|---|---|---|---|---|

```
#  ****************************
#  **                        **
#  **  Congratulations !!    **
#  **                        **       |__||
#  **  Simulation PASS !!    **      / 0.0  |
#  **                        **    /_____   |
#  **  Your score =100       **   /^ ^ ^ \  |
#  **                        **  |^ ^ ^ ^ |w|
#  ****************************   \m___m__|_|
# ** Note: $finish    : testfixture.v(157)
#    Time: 4422500 ps  Iteration: 0  Instance: /test
# 1
# Break in Module test at testfixture.v line 157
```

```
#
#  ****************************
#  **                        **
#  **  Congratulations !!    **
#  **                        **       |__||
#  **  Simulation PASS !!    **      / 0.0  |
#  **                        **    /_____   |
#  **  Your score =100       **   /^ ^ ^ \  |
#  **                        **  |^ ^ ^ ^ |w|
#  ****************************   \m___m__|_|
# ** Note: $finish    : testfixture.v(157)
#    Time: 4430061 ps  Iteration: 0  Instance: /test
# 1
# Break in Module test at testfixture.v line 157
```

## Synthesis Result

| Total logic elements | 2987 |
|---|---|
| Total memory bit | 0 |
| Embedded multiplier 9-bit element | 0 |

**Flow Summary**

<<Filter>>

| | |
|---|---|
| Flow Status | Successful - Sat May 25 10:58:25 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 | 2,987 / 55,856 ( 5 % ) |
| Total registers | 497 |
| 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

**於 state INPUT_DATA 讀取 testbench 傳送過來的 data，將之依序存入數組 values 中。為了建立一個 Max Heap，將數組內索引由大到小依序與一個 complete binary tree 對應，數組最後的索引利用：(idx-1)/2 即可得到 the last parent Node。於 state COMPARE_LEFT_CHILD 開始， 由 the last parent 到 Root 採取 bottom up 建立 Max Heap。於 state GET_COMMAND 信號為 cmd_valid 獲取指定的 command：**

**1. Extract Max：將 the last node 取代 root node，並且對 root node 於 state COMPARE_LEFT_CHILD 採用 bottom up。**

**2. Increase_Value：對指定索引值進行更新，於 state UPDATE_VALUE 迭代比對 Parent Node 大小再判斷是否進行 SWAP。**

**3. Insert_Data：新加入的數值，，於 state UPDATE_VALUE 迭代比對 Parent Node 大小再判斷是否進行 SWAP。**

**4. Write：將整理完成的數組 values，每一 cycle 依序將索引及數值傳回 testbench。**

***Scoring = 2987 * 4430.061 = 13232592.207***

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