2024 Digital IC Design
Homework 4: Max-Priority Queue

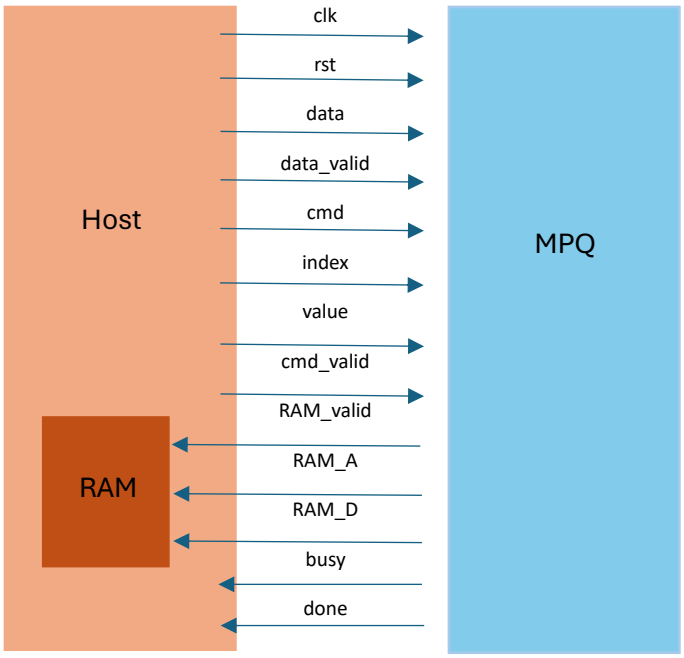
1. Introduction:

Please complete the circuit design of the Max-Priority Queue. This circuit is capable of reading data and performing the functions of Build_Queue, Extract_Max, Increase_Value, and Insert_Data according to specified control commands. Finally, the results will be written to RAM.

Details regarding the signal definitions and the operation of the MPQ are described below.

2. Specification :

2.1 Block Overview



2.2 I/O Interface

Signal	I/O	Bit Width	Description
clk	input	1	This circuit is a synchronous design triggered at the positive edge of *clk*.
rst	input	1	Active-high asynchronous reset signal.
data	input	8	Original Data: When "data_valid" is high, it indicates that the data is valid.
data_valid	Input	1	When this signal is high, it indicates that the data input is valid.
cmd	input	3	Command Input Signals: There are five types of command inputs for this controller. Command inputs are valid only when "cmd_valid" is high and "busy" is low.
Index	input	8	Instruction-Related Signals, detailed below.
Value	input	8	Instruction-Related Signals, detailed below.
cmd_valid	input	1	When this signal is high, it indicates that the cmd instruction is a valid input command.
RAM_valid	output	1	RAM Memory Data Enable Signal: When it is high, it indicates that the data and address buses transmitted from the MPQ end are valid.
RAM_A	output	8	RAM Address Bus: The MPQ end needs to use this bus to instruct the Host end's Image RAM memory to write data to the specified address.
RAM_D	output	8	RAM Data Bus: The MPQ end utilizes this bus to write data to the RAM memory module on the Host end.
busy	output	1	System Busy Signal: Description: When this signal is high, it indicates that the controller is executing the current command and cannot accept any new command inputs. When this signal is low, the system is ready to accept new commands. Upon reset, the default setting is high.
done	output	1	When the controller completes writing to RAM, setting "done" to high indicates completion.

2.3 File Description

This assignment will only provide examples for test case 0 (Build, Write). You will need to perform software verification for the remaining test cases 1 to 3. Test case 1 requires adding the Extract_Max function, test case 2 requires adding the Increase_Value function, and test case 3 requires adding the Insert_Data function. The folder location for test case 0 is "/dat/P0", for test case 1 is "/dat/P1", and so on.

Note : Max queue size is 255,data range is 0~255.

File Name	Description
MQP.v	The module of MPQ, which is the top module in this design.
tb.v	The testbench file.
pat.dat	Input Raw Data: The data format is hexadecimal.
cmd.dat	Command Data: The data format is binary.
index.dat	Index Data: The data format is hexadecimal.
value.dat	Value Data: The data format is hexadecimal.
golden.dat	Golden data for MPQ verification.

2.4 Function Description

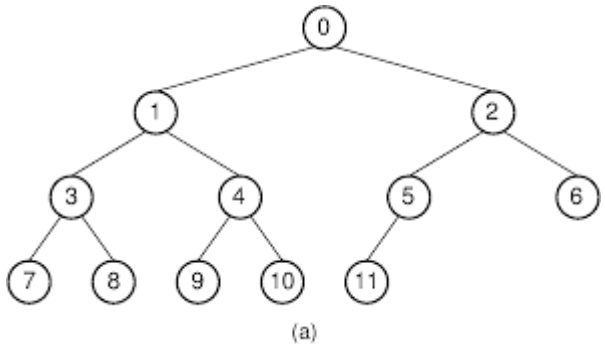
2.4.1 CMD description

Description	CMD	index	value
Build_Queue	0 (000)	0 (unused)	0 (unused)
Extract_Max	1 (001)	0 (unused)	0 (unused)
Increase_Value	2 (010)	index	value
Insert_Data	3 (011)	0 (unused)	value
Write	4 (100)	0 (unused)	0 (unused)

Each instruction should be referenced according to the pseudo code provided in the reference.

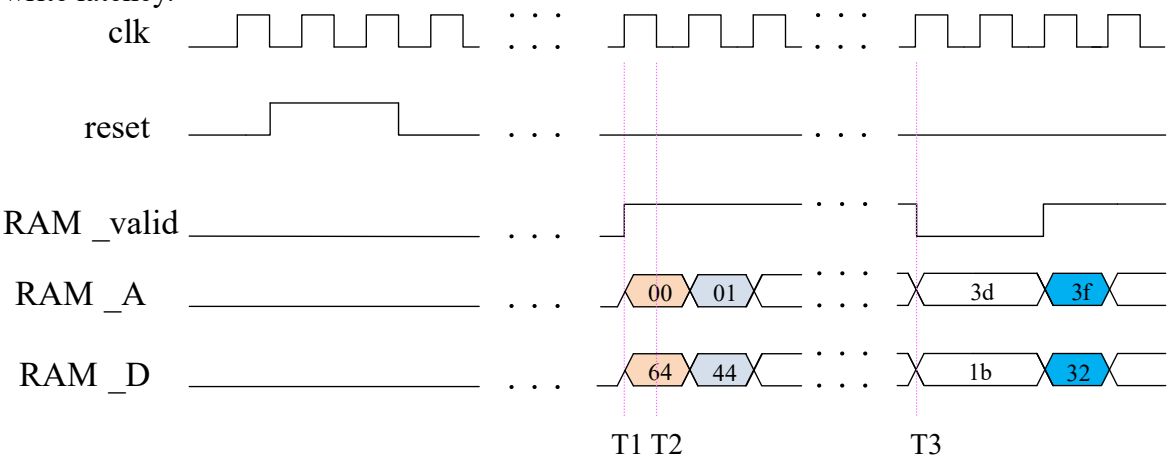
2.4.2 RAM Correspondence and Timing Specifications

The result computed by the MPQ is a complete tree. The correspondence is as illustrated in the following diagram:



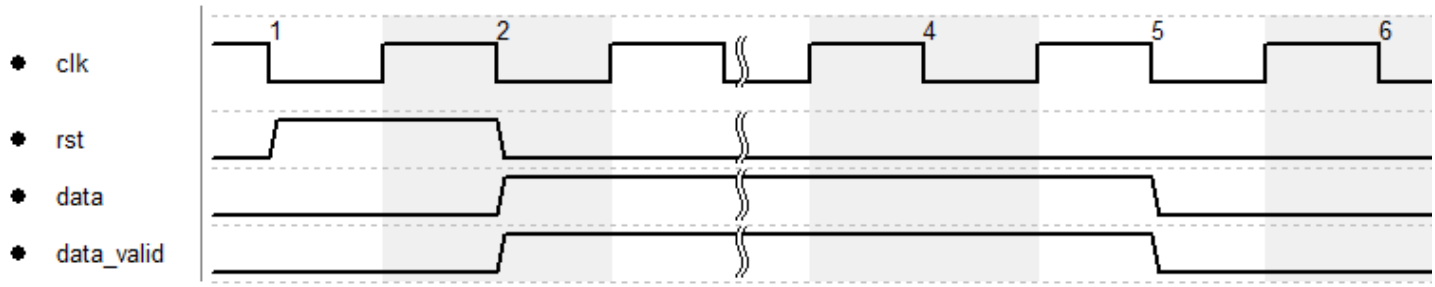
(a)

The timing of RAM output is as shown in Figure. If the RAM_valid signal is high (as indicated at time T1 in the Figure), the RAM will simultaneously place the desired address on the RAM_A bus and the data on the RAM_D bus at each negative edge of the clock signal. At the negative edge of the HOST clock signal at time T2, the system will perform a writing operation. When writing data, keep RAM_valid high and update RAM_A and RAM_D as necessary. To end the data write, set RAM_valid to low at time T3. This memory does not need to consider read and write latency.

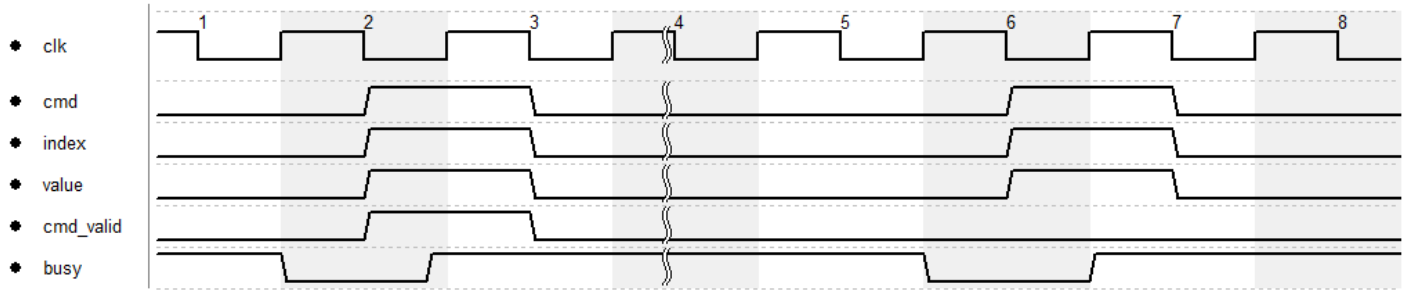


2.5 Timing Specification Diagram

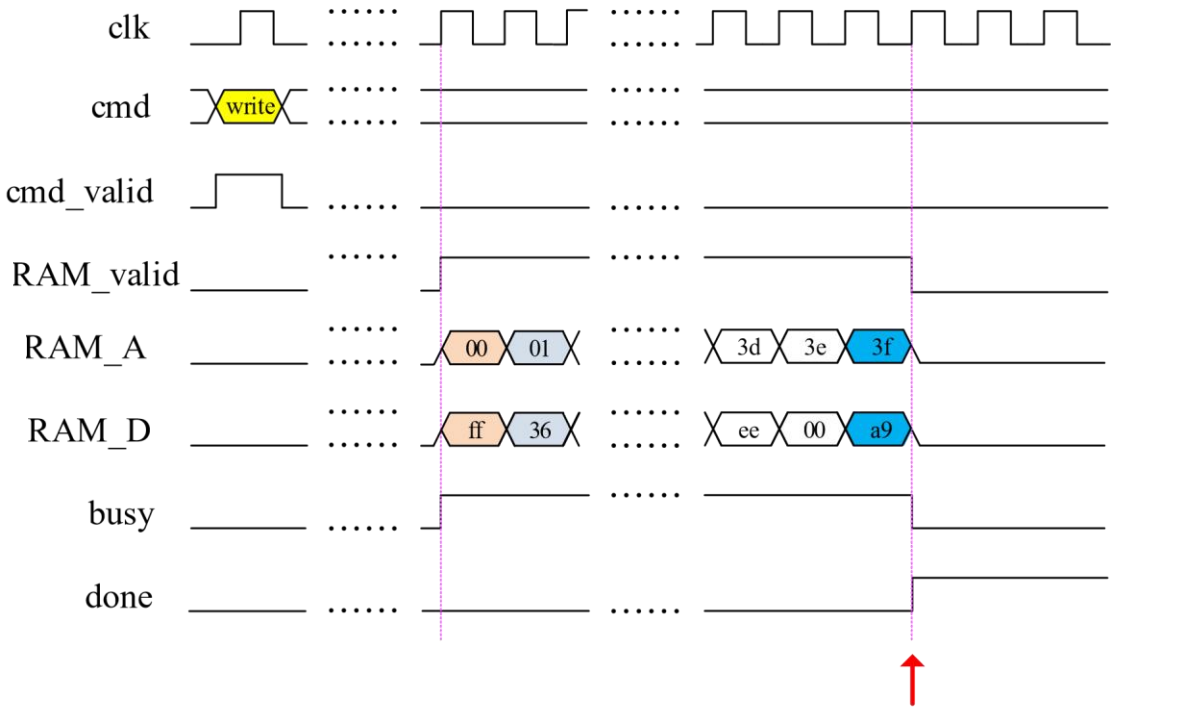
- The timing specification diagram after reset is shown in Figure.
  - After the circuit is reset, the controller will output n pieces of data.
  - During the entire process, the 'busy' signal stays high, showing it's not ready for commands.



- The timing specification diagram for control commands (Build_Queue, Extract_Max, Increase_Value, Insert_Data) is as shown in the following figure.
  - Throughout the entire processing, the "busy" signal remains high. After the operation is completed, "busy" is set back to low to accept new command inputs.

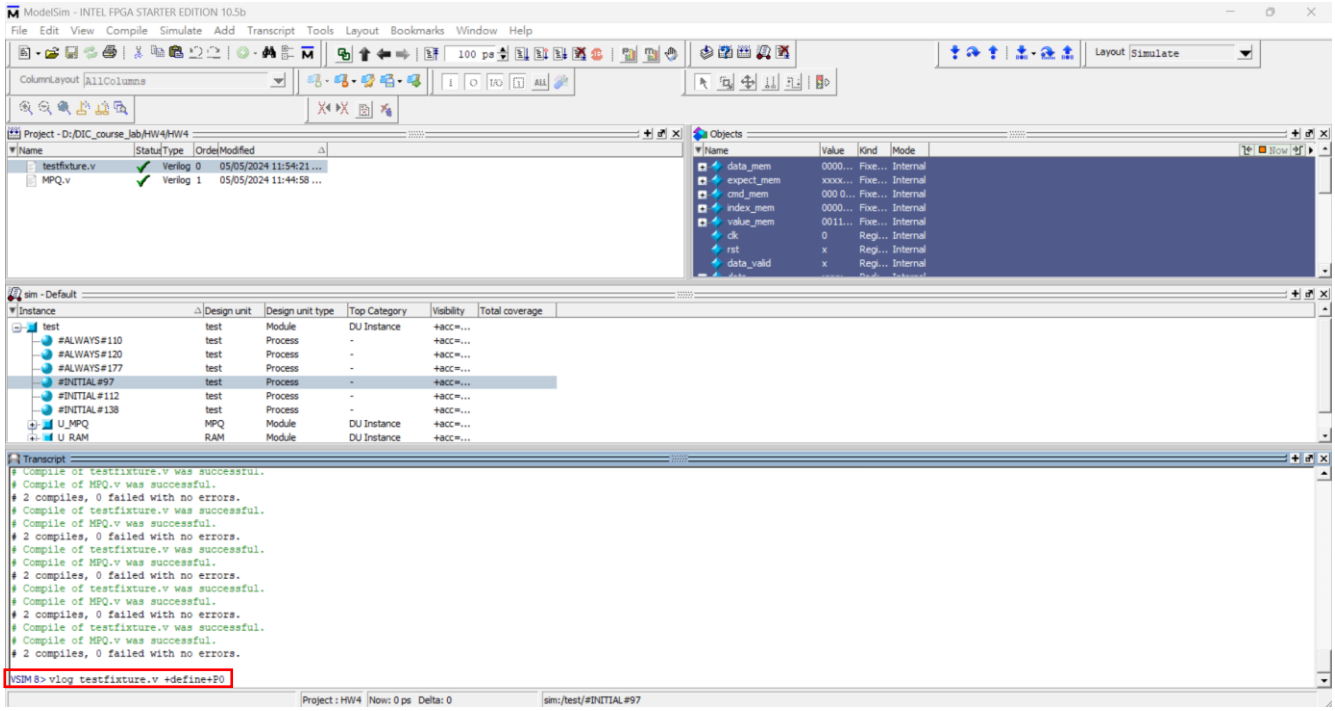


- The timing specification diagram for the write command is as shown in the following figure.
  - When executing the write command, the controller will write the processed image data into the RAM.
  - When RAM_valid is high, it indicates writing to RAM. At this point, address signals can be input to write image data into the RAM.
  - After writing is completed, the "done" signal is set to high to indicate completion. At this point, the testfixture will compare the written data in RAM with the golden pattern.



3. Functional Simulation

You should type “vlog testfixture.v +define+PX” on modelsim Transcript to compile testfixture.v to test Pattern X.



4. Gate-Level Simulation

4.1 Synthesis

Your code should be synthesizable. After it is synthesized in Quartus, files named MM.vo and MM_v.sdo will be obtained.

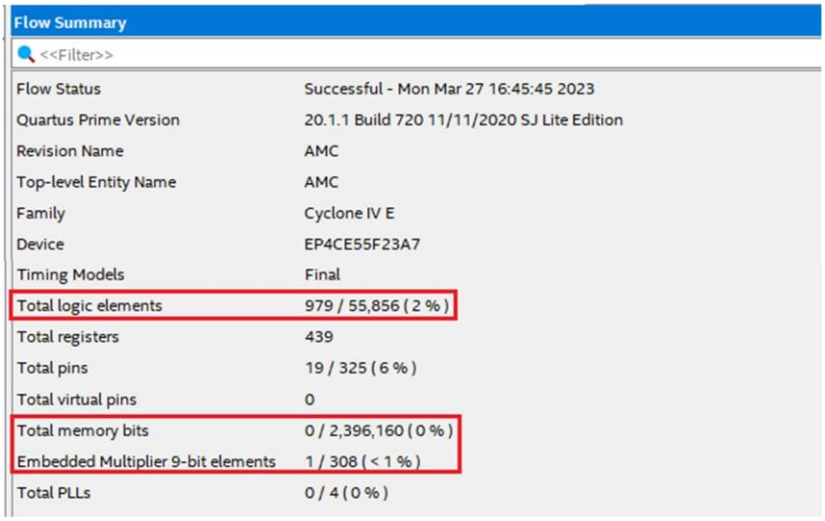
DEVICE：Cyclone IV E - EP4CE55F23A7

4.2 Simulation

All of the results should be generated correctly using MM.vo and MM_v.sdo, and you will get the following message in ModelSim simulation.(score = 0.2 * total_score)

5. Performance

The performance is scored by the total logic elements, total memory bit, and embedded multiplier 9-bit element your design used in gate-level simulation and the simulation time your design takes.



Flow Summary	
<<Filter>>	
Flow Status	Successful - Mon Mar 27 16:45:45 2023
Quartus Prime Version	20.1.1 Build 720 11/11/2020 SJ Lite Edition
Revision Name	AMC
Top-level Entity Name	AMC
Family	Cyclone IV E
Device	EP4CE55F23A7
Timing Models	Final
Total logic elements	979 / 55,856 (2 %)
Total registers	439
Total pins	19 / 325 (6 %)
Total virtual pins	0
Total memory bits	0 / 2,396,160 (0 %)
Embedded Multiplier 9-bit elements	1 / 308 (< 1 %)
Total PLLs	0 / 4 (0 %)

The performance score will be decided by your ranking in all received homework. Only designs that passed gate-level simulation and meet resource limitations will be considered in the ranking. Otherwise, you can't get performance score.

The scoring standard: (The smaller, the better)

Scoring = Area cost * Timing cost

Area cost = Total logic elements + total memory bits + 9*embedded multiplier 9-bit elements

Timing cost = Total cycle used*clock width

6.Scoring

1.Functional simulation [60%]

2.Gate-Level simulation [20%]

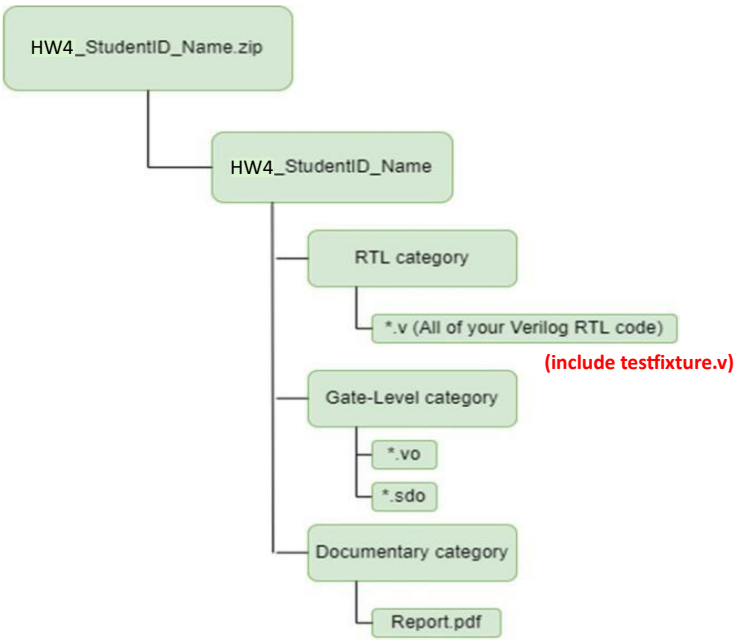
3.Performane [20%]

7. Submission

7.1. Submitted files

You should classify your files into three directories and compress them to .zipformat. The naming rule is HW4_studentID_name.zip. If your file is not named according to the naming rule, you will lose five points.

	RTL category
*.v	All of your Verilog RTL code **(include testfixture.v)**
	Gate-Level category
*.vo	Gate-Level netlist generated by Quartus
*.sdo	SDF timing information generated by Quartus
	Documentary category
*.pdf	The report file of your design (in pdf).



Note

In this homework, you are allowed to modify the defined *CYCLE* and *End_CYCLE* in testbench file. 'CYCLE' decides the clock width that is used to validate your design,and 'End_CYCLE' decides the maximum cycles your circuit takes to complete the simulation. Please do not modify any other content of the testbench.

Please submit your .zip file to folder HW4 in moodle.

Deadline: 2024/05/26 23:55

If you have any problem, please contact TA by email
m16124114@gs.ncku.edu.tw