

IC Design

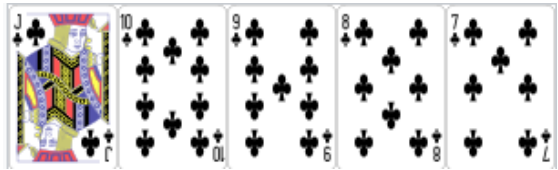
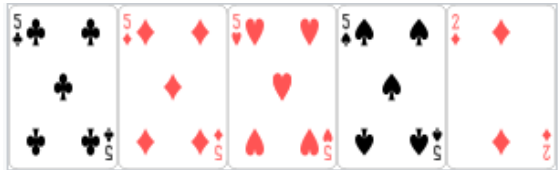


Homework # 3



(Due on 2023/11/23, 09:10. Verilog code and Report upload to NTUCool)

- ✧ Plagiarism is not allowed. 10% penalty for each day of delay.
- ✧ Any further questions, you can send e-mail to the TA (陳帝宇) or leave messages on the board of the class website.
- ✧ TA email: tp62u4m3@gmail.com, EE2-329

Specifications

In this homework, you are asked to design a **gate-level combinational circuit** that finds the type of the given hands of five given playing cards. The inputs of this circuit are **five 6-bit** digital values, denoted as $i0, i1, i2, i3, i4$. **The 2 MSBs** in the 6-bits **indicate their suits**. 0, 1, 2, and 3 refer to spades, hearts, diamonds, and clubs, respectively. While **the remaining 4 bits are their ranks**. Ace, Jack, Queen, and King are represented by 1, 11, 12, and 13, respectively. The output of the circuit, denoted as $type$, is a **4-bit unsigned number**. The possible types are listed below:

Types and the Corresponding Output	Descriptions	Example
Straight Flush 8	A Straight of the same suit	
Four of a Kind 7	4 of the 5 cards have identical ranks	
Full House 6	3 cards with identical ranks accompanied by a pair with another rank	
Flush 5	5 cards that don't form a straight of the same suit	

Straight 4	5 cards with sequential ranks, not all of the same suit Ace cannot appear in the middle	
Three of a Kind 3	3 cards with identical ranks	
Two Pair 2	2 cards of one rank, and 2 cards of another rank	
One Pair 1	2 cards with identical ranks	
High Card 0	None of the above	

Note that **Ace cannot be in the middle of a Straight**, i.e. K-A-2-3-4 is not a Straight (nor Straight Flush) and is seen as a High Card (or Flush).

Below are some examples of the I/O:

Input					Output
<i>i0</i>	<i>i1</i>	<i>i2</i>	<i>i3</i>	<i>i4</i>	<i>type</i>
0x29	0x0c	0x0d	0x26	0x15	0x0
0x37	0x1a	0x05	0x27	0x15	0x2
0x2d	0x1c	0x2a	0x11	0x0b	0x4
0x33	0x3d	0x31	0x38	0x39	0x5
0x1b	0x28	0x0b	0x3b	0x18	0x6

[**HINT**] You can find the relation between all possible pairs of cards first (same rank, same suit, differ by 1, etc.). If you have trouble identifying straights, brute force is an acceptable approach.

There are some important things that you should notice:

- Your design should be based on the **standard cells in the lib.v**. All logic operations in your design **MUST consist of the standard cells** instead of using

the operands such as “+”, “-”, “&”, “|”, “>”, and “<”. **Note that the score of HW3 will be 0 if you use any of them.**

- Design your homework in the given “poker.v” file. **You are NOT ALLOWED to change the filename and the header of the top module (i.e., the module name and the I/O ports).**
- If your design contains more than one module, don’t create a new file for them. Just put those modules in “poker.v.”
- The output waveform will be dumped to file “**poker.fsdb.**” You can use nWave to examine it.
- **For each set of input data, the test bench will allow your circuit to calculate the sorter outputs within 20ns.** Once exceeding 20ns or detecting the correct answer from your circuit, the test bench will soon provide the new data set to your design until all 10000 data sets have been simulated.

Grading

1. Gate-level design using Verilog (80%)

Your score will depend on both the correctness and performance of your design. We provide a “public” test bench with 10000 datasets. What follows is the grading policy:

Correctness & Performance	Score
Fail to pass the test bench.	40 * (1-err #/10000)
Functionally correct	40
Critical Path < 7ns	45
Critical Path < 6ns	50
Critical Path < 5ns	55
Critical Path < 4ns	60
Critical Path < 3ns	70
Critical Path < 2.5ns	80
Using operands, not standard cell logic	0
Plagiarism	0

Testbench will provide related information for grading:

```
Congratulations! Your critical path is below 2.5!
$finish called from file "tb_poker.v", line 267.
$finish at simulation time          210000000
      V C S   S i m u l a t i o n   R e p o r t
Time: 210000000 ps
CPU Time:      1.430 seconds;      Data structure size:  0.2Mb
Critical Path <2.5ns
```

2. Report (20%)

You should also describe and discuss your design. Below are the required items in your report.

➤ Circuit diagram (10%)

Plot the circuit diagram of your design. You are encouraged to plot it hierarchically so that the reader can understand your design easily. If you simplify your sub-circuit with a K-map, you can also show your K-map and logic expression instead of the circuit diagram of the sub-circuit for better understanding.

➤ Discussion (10%)

Discuss about your design. For example, how do you identify the straights, which subtractor technique is adopted in your design, and how do you improve your critical path.

Notification

➤ Following are the files you will need (available on the class website)

HW3.zip includes

- **HW3_2023.pdf** : this document.
- **HW3_tutorial.pdf** Verilog introduction

- **poker.v:**

Dummy design file. Program the design in this file.

The header of the top module and the declaration of the I/O ports are predefined in this file, and you cannot change them.

- **lib.v:** standard cells.
- **tb_poker.v, tb_poker_pattern.v:**

Testbench for your design.

- **i0.dat, i1.dat, i2.dat, i3.dat, i4.dat:**

Input patterns for the test bench. Put these files in the folder that contains **tb_poker.v** when doing the simulation.

- **golden.dat:**

Output patterns of correct answers for the test bench. Put this file in the folder that contains **tb_poker.v** when doing the simulation.

➤ The following files should be compressed and uploaded to CEIBA by due time.

- Report (PDF format)
- poker.v

➤ File name rule : *HW3_(student id)_v#*

Ex. HW3_b05901058_v1.zip

Ex. HW3_b05901058_v2.zip

TA：陳帝宇，EE2-329

TA email: tp62u4m3@gmail.com

HW3 Office hours: 11/21 18:30-20:30 @ 電二 329 室

11/22 18:30-20:30 @ 電二 329 室

If you have no time during office hours, you can email TA to discuss another time for appointment.