

IC Design HW3

B10901176 蔡弘祥

1. Circuit Overview

Some Checkers	$\bar{1}0[5:4]$	$\bar{1}n0[1:0]$	flush checker	out	flush
	$\bar{1}1[5:4]$	$\bar{1}n1[1:0]$			flush'
	$\bar{1}2[5:4]$	$\bar{1}n2[1:0]$			
	$\bar{1}3[5:4]$	$\bar{1}n3[1:0]$			
	$\bar{1}4[5:4]$	$\bar{1}n4[1:0]$			
	$\bar{1}0[3:0]$	$\bar{1}n0[3:0]$			
Some Checkers	$\bar{1}1[3:0]$	$\bar{1}n1[3:0]$	straight checker	out	Straight
	$\bar{1}2[3:0]$	$\bar{1}n2[3:0]$			straight'
	$\bar{1}3[3:0]$	$\bar{1}n3[3:0]$			
	$\bar{1}4[3:0]$	$\bar{1}n4[3:0]$			
	$\bar{1}0[3:0]$	$\bar{1}n0[3:0]$			
	$\bar{1}1[3:0]$	$\bar{1}n1[3:0]$			
Some Checkers	$\bar{1}2[3:0]$	$\bar{1}n2[3:0]$	4 of a kind checker	out	4ofakind
	$\bar{1}3[3:0]$	$\bar{1}n3[3:0]$			4ofakind'
	$\bar{1}4[3:0]$	$\bar{1}n4[3:0]$			
	$\bar{1}0[3:0]$	$\bar{1}n0[3:0]$			
	$\bar{1}1[3:0]$	$\bar{1}n1[3:0]$			
	$\bar{1}2[3:0]$	$\bar{1}n2[3:0]$			
Some Checkers	$\bar{1}3[3:0]$	$\bar{1}n3[3:0]$	full house checker	out	full house
	$\bar{1}4[3:0]$	$\bar{1}n4[3:0]$			full house'
	$\bar{1}0[3:0]$	$\bar{1}n0[3:0]$			
	$\bar{1}1[3:0]$	$\bar{1}n1[3:0]$			
	$\bar{1}2[3:0]$	$\bar{1}n2[3:0]$			
	$\bar{1}3[3:0]$	$\bar{1}n3[3:0]$			
Some Checkers	$\bar{1}0[3:0]$	$\bar{1}n0[3:0]$	3 of a kind possible checker	out	3ofakindpossible
	$\bar{1}1[3:0]$	$\bar{1}n1[3:0]$			3ofakindpossible'
	$\bar{1}2[3:0]$	$\bar{1}n2[3:0]$			
	$\bar{1}3[3:0]$	$\bar{1}n3[3:0]$			
	$\bar{1}4[3:0]$	$\bar{1}n4[3:0]$			
	$\bar{1}0[3:0]$	$\bar{1}n0[3:0]$			
Some Checkers	$\bar{1}1[3:0]$	$\bar{1}n1[3:0]$	2 pairs possible checker	out	2pairspossible
	$\bar{1}2[3:0]$	$\bar{1}n2[3:0]$			2pairspossible'
	$\bar{1}3[3:0]$	$\bar{1}n3[3:0]$			
	$\bar{1}4[3:0]$	$\bar{1}n4[3:0]$			
	$\bar{1}0[3:0]$	$\bar{1}n0[3:0]$			
	$\bar{1}1[3:0]$	$\bar{1}n1[3:0]$			
Some Checkers	$\bar{1}2[3:0]$	$\bar{1}n2[3:0]$	1 pair possible checker	out	1pairpossible
	$\bar{1}3[3:0]$	$\bar{1}n3[3:0]$			1pairpossible'
	$\bar{1}4[3:0]$	$\bar{1}n4[3:0]$			
	$\bar{1}0[3:0]$	$\bar{1}n0[3:0]$			
	$\bar{1}1[3:0]$	$\bar{1}n1[3:0]$			
	$\bar{1}2[3:0]$	$\bar{1}n2[3:0]$			

Type[3]	
Type[2]	
Type[1]	
Type[0]	

2. Temp[2:0]

Explanation:

For those combinations which are not straight, flush, or straight flush, we first determine the output named temp[2:0] using some checkers mentioned above.

Checker Name	Abbreviation
4 of a kind	4O
Full house	FH
3 of a kind possible	3O
2 pairs possible	2P
1 pair possible	1P

	$4O=0$	$2P, 1P$	$4O=1$	$2P, 1P$
	00 01 11 10		00 01 11 10	
	00 01 11 10	0 1 2 X	00 01 11 10	7 7 7 7
	FH,30 01	X 3 X X	FH,30 01	7 7 7 7
temp[2:0]	11 X X 6 X		11 X X X X	
	10 X X X X		10 X X X X	

	$4O=0$	$2P, 1P$	$4O=1$	$2P, 1P$
	00 01 11 10		00 01 11 10	
	00 01 11 10	0 0 0 X	00 01 11 10	1 1 1 1
	FH,30 01	X 0 X X	FH,30 01	1 1 1 1
temp[2]	11 X X X		11 X X X X	
	10 X X X X		10 X X X X	

$\text{temp}[2] = 4O + FH = (4O'FH')$ '

	$4O=0$	$2P, 1P$	$4O=1$	$2P, 1P$
	00 01 11 10		00 01 11 10	
	00 01 11 10	0 0 X	00 01 11 10	1 1 1
	FH,30 01	X X X	FH,30 01	1 1 1
temp[1]	11 X X X		11 X X X X	
	10 X X X X		10 X X X X	

$\text{temp}[1] = 4O + 30 + 2P = (4O'30'2P')$ '

	$4O=0$	$2P, 1P$	$4O=1$	$2P, 1P$
	00 01 11 10		00 01 11 10	
	00 01 11 10	0 0 X	00 01 11 10	1 1 1
	FH,30 01	X X X	FH,30 01	1 1 1
temp[0]	11 X 0 X		11 X X X X	
	10 X X X X		10 X X X X	

$\text{temp}[0] = 4O + 2P'1P = [4O'(2P'1P)]'$

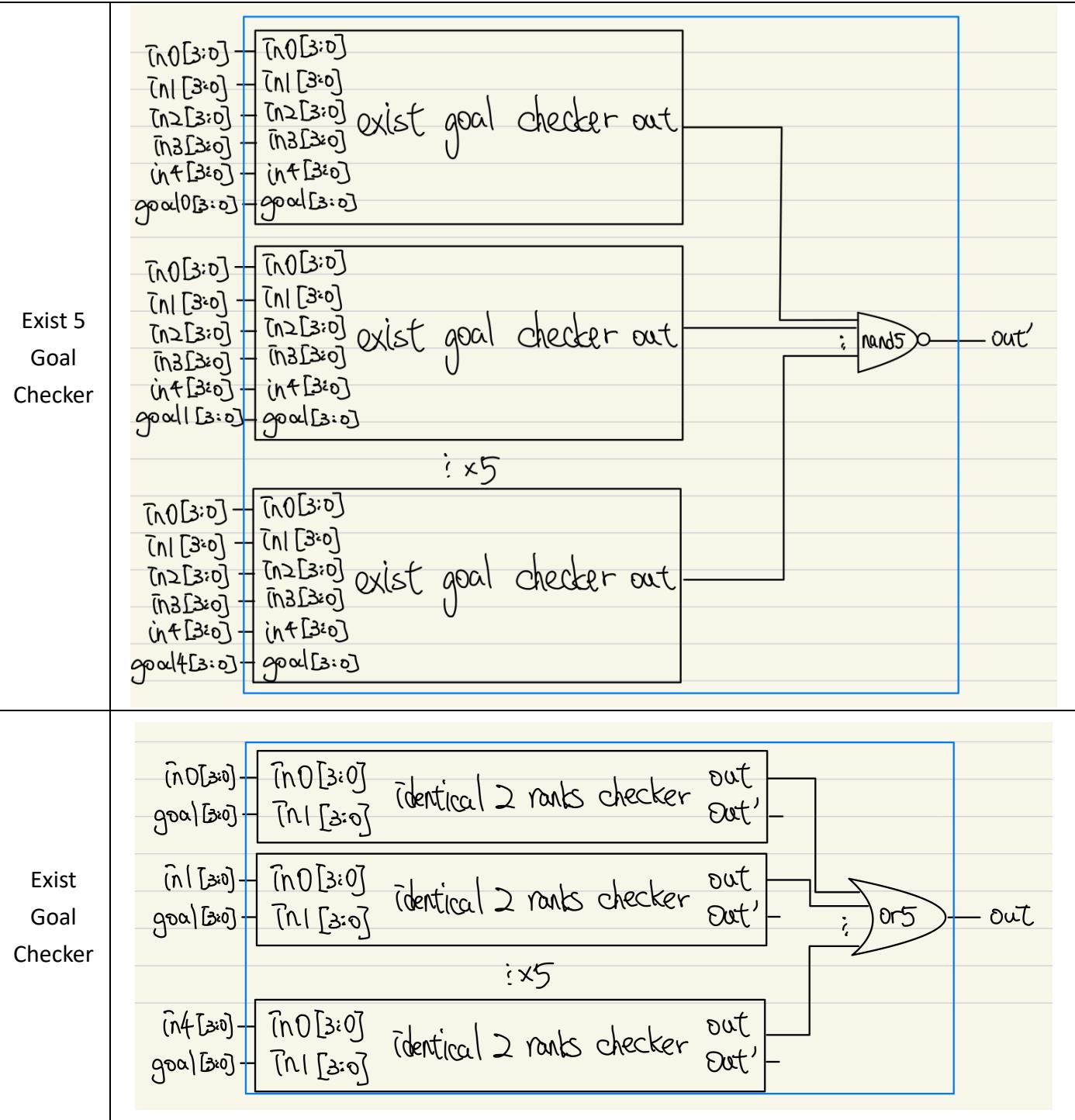
3. Self-defined Sub Module Overview

Sub Module Name	input	output	Explanation
Flush Checker	in0[1:0] in1[1:0] in2[1:0] in3[1:0] in4[1:0]	out out'	Input the first two bits of all cards out = true if they are flush or not
Straight Checker	in0[3:0] in1[3:0] in2[3:0] in3[3:0] in4[3:0]	out out'	Input the last four bits of all cards out = true if they are straight or not
Exist 5 Goal Checker	goal0[3:0] goal1[3:0] goal2[3:0] goal3[3:0] goal4[3:0] in0[3:0] in1[3:0] in2[3:0] in3[3:0] in4[3:0]	out'	Input the five numbers from the cards and the five goal numbers out = true if all goal numbers exist in the cards
Exist Goal Checker	goal[3:0] in0[3:0] in1[3:0] in2[3:0] in3[3:0] in4[3:0]	out	Input the five numbers from the cards and a goal number out = true if the goal number exists in the cards
4 of a Kind Checker	in0[3:0] in1[3:0] in2[3:0] in3[3:0] in4[3:0]	out out'	Input the last four bits of all cards out = true if they are 4 of a kind or not
Full House Checker	in0[3:0] in1[3:0] in2[3:0] in3[3:0] in4[3:0]	out out'	Input the last four bits of all cards out = true if they are full house or not
3 of a kind Possible Checker	in0[3:0] in1[3:0]	out out'	Input the last four bits of all cards out = true if there are 3 cards with the same rank

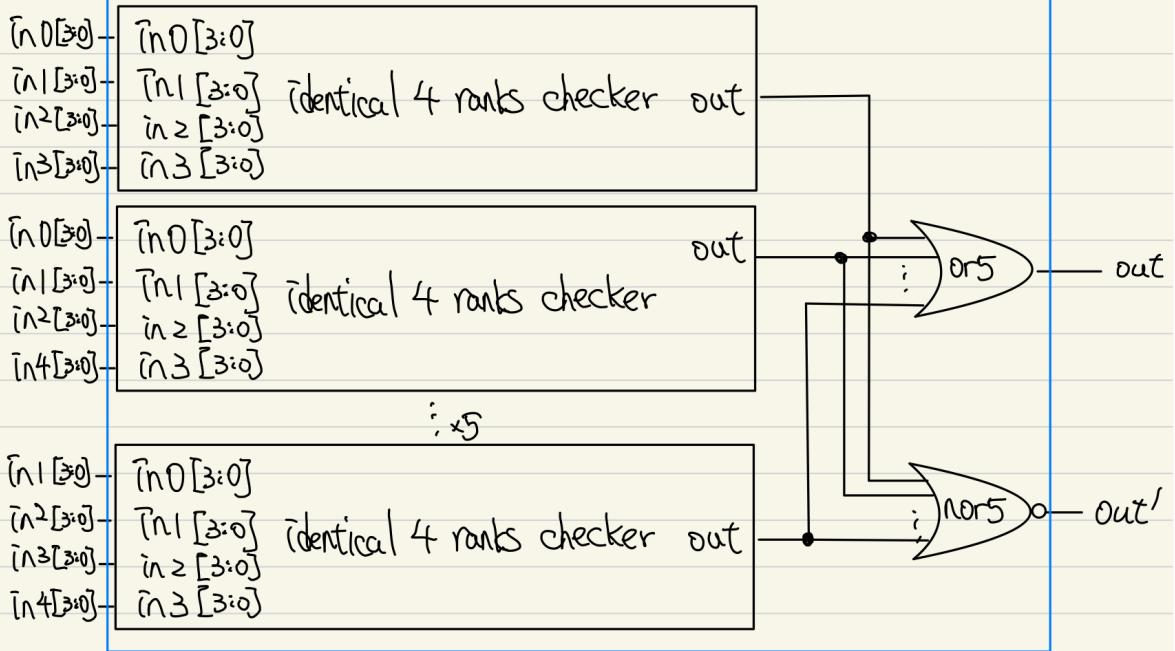
	in2[3:0] in3[3:0] in4[3:0]		
2 Pairs Possible Checker	in0[3:0] in1[3:0] in2[3:0] in3[3:0] in4[3:0]	out out'	Input the last four bits of all cards out = true if there are 2 pairs or 4 of a kind
2 Pairs Possible Sub Checker	in0[3:0] in1[3:0] in2[3:0] in3[3:0]	out out'	Input the four numbers from the cards out = true if there are at least two pairs among the cards (all same rank). All 4 cards with the same rank will also return true.
1 Pair Possible Checker	in0[3:0] in1[3:0] in2[3:0] in3[3:0] in4[3:0]	out out'	Input the last four bits of all cards out = true if there are at least a pair among the cards
Identical 4 Ranks Checker	in0[3:0] in1[3:0] in2[3:0] in3[3:0]	out	Input four numbers out = true if they are the same
Identical 3 Ranks Checker	in0[3:0] in1[3:0] in2[3:0]	out	Input three numbers out = true if they are the same
Identical 2 Ranks Checker	in0[3:0] in1[3:0]	out out'	Input two numbers out = true if they are the same
Same 5 Bits Checker	in0 in1 in2 in3 in4	out	Input five bits out = true if they are the same
Same 4 Bits Checker	in0 in1 in2 in3	out	Input four bits out = true if they are the same
Same 3 Bits Checker	in0 in1 in2	out	Input three bits out = true if they are the same
MUX4C	A,B,C,D CTRL1 CTRL2	out	Cascading Muxes, CTRL2 should be the slower signal

4. Sub Module Circuit Diagram

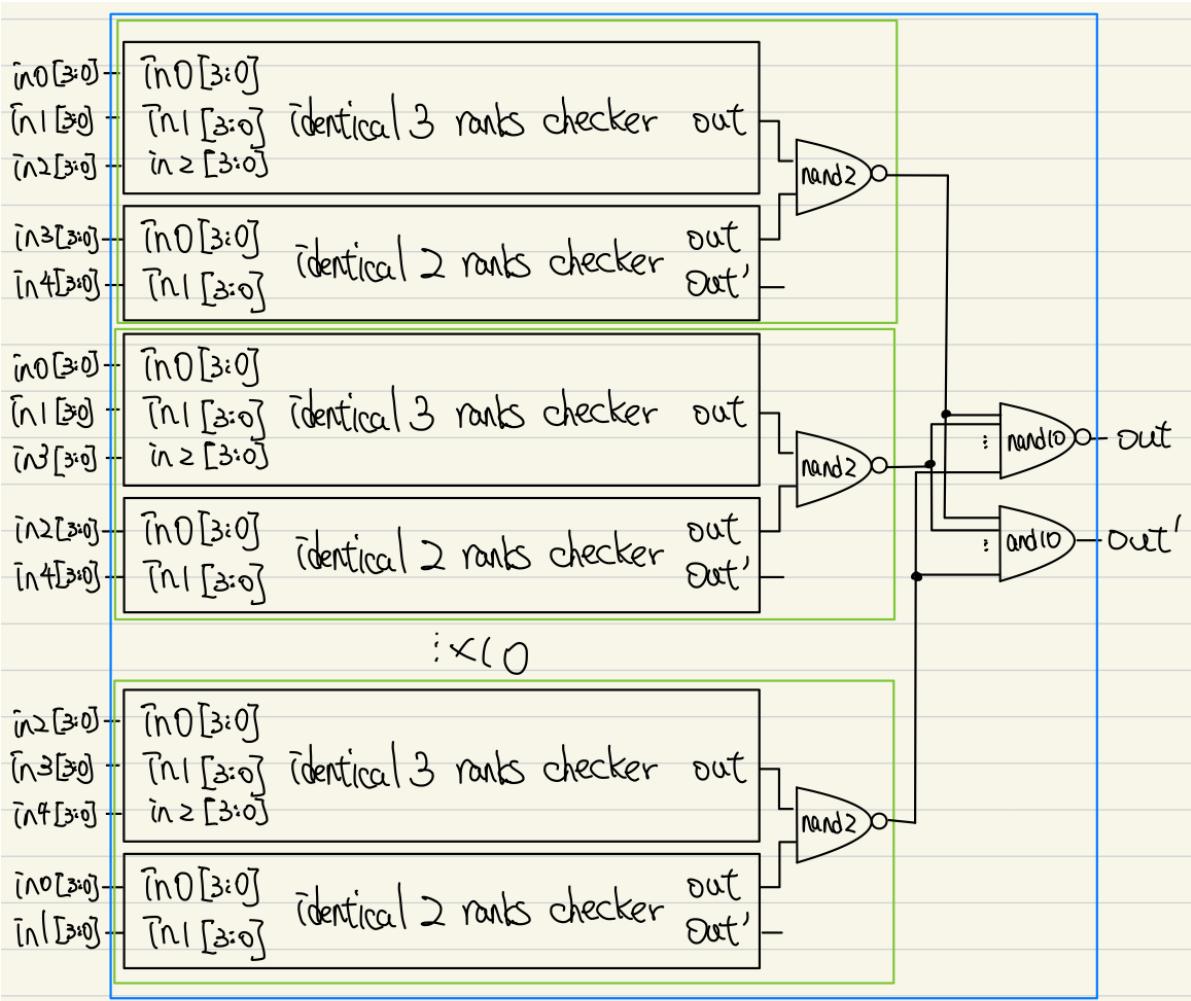
Sub Module Name	Diagram
Flush Checker	<p>Same 5 bits checker</p>
Straight Checker	<p>exist 5 goal checker</p> <p style="text-align: center;">; x 10</p>



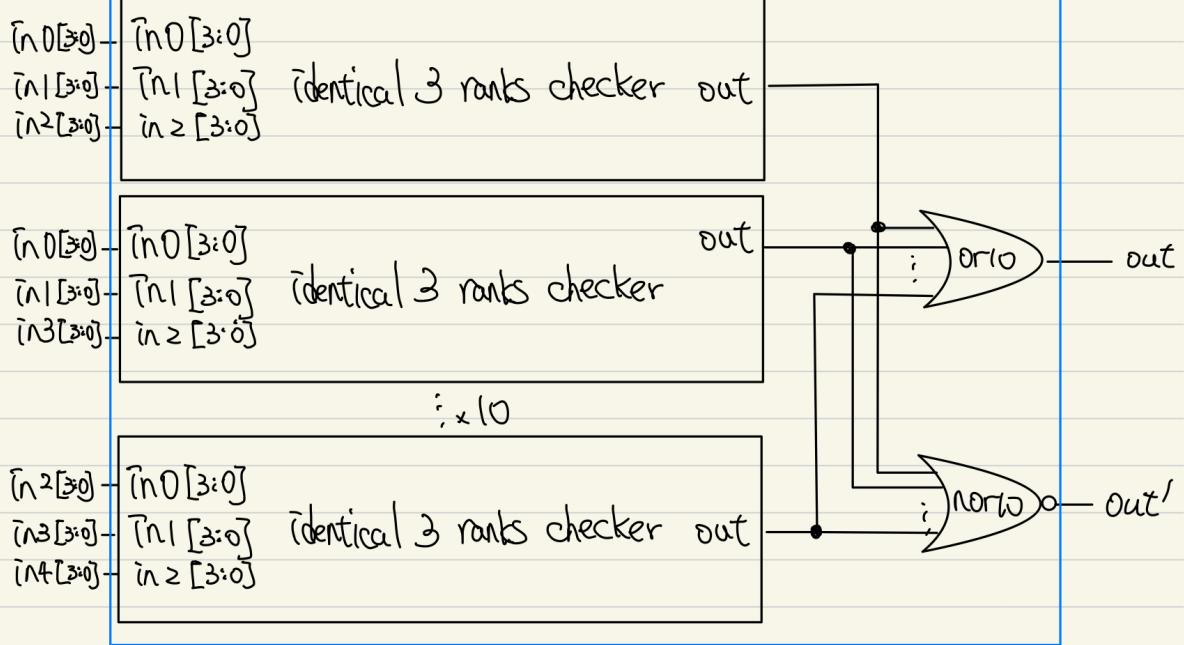
4 of a Kind Checker



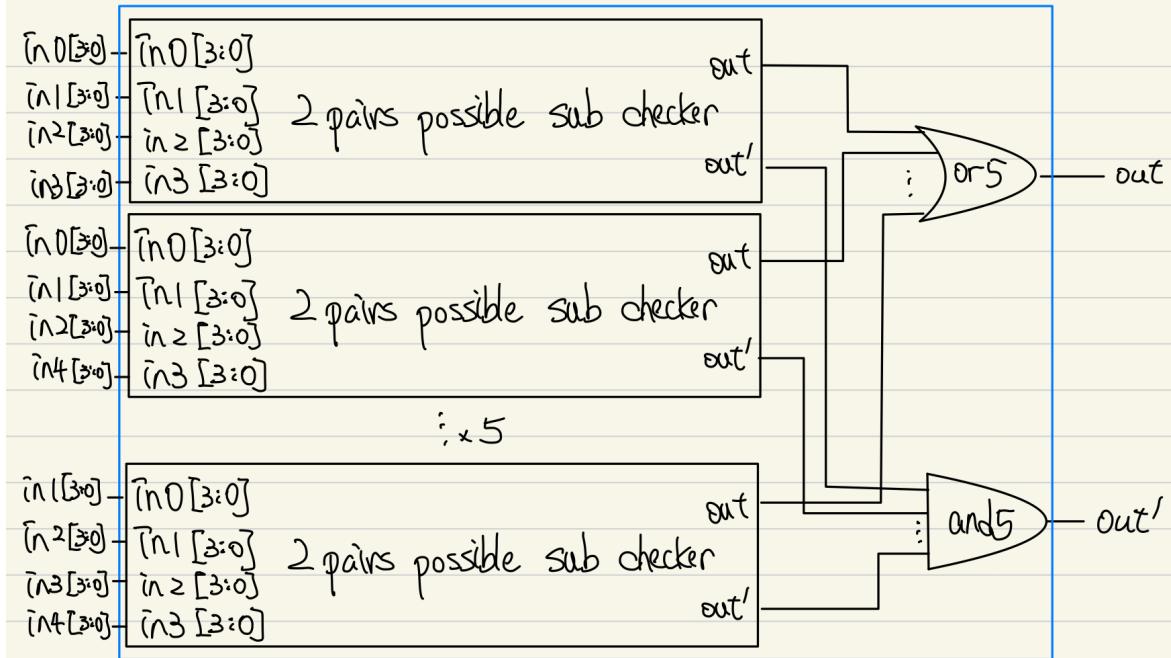
Full House Checker



3 of a kind Possible Checker

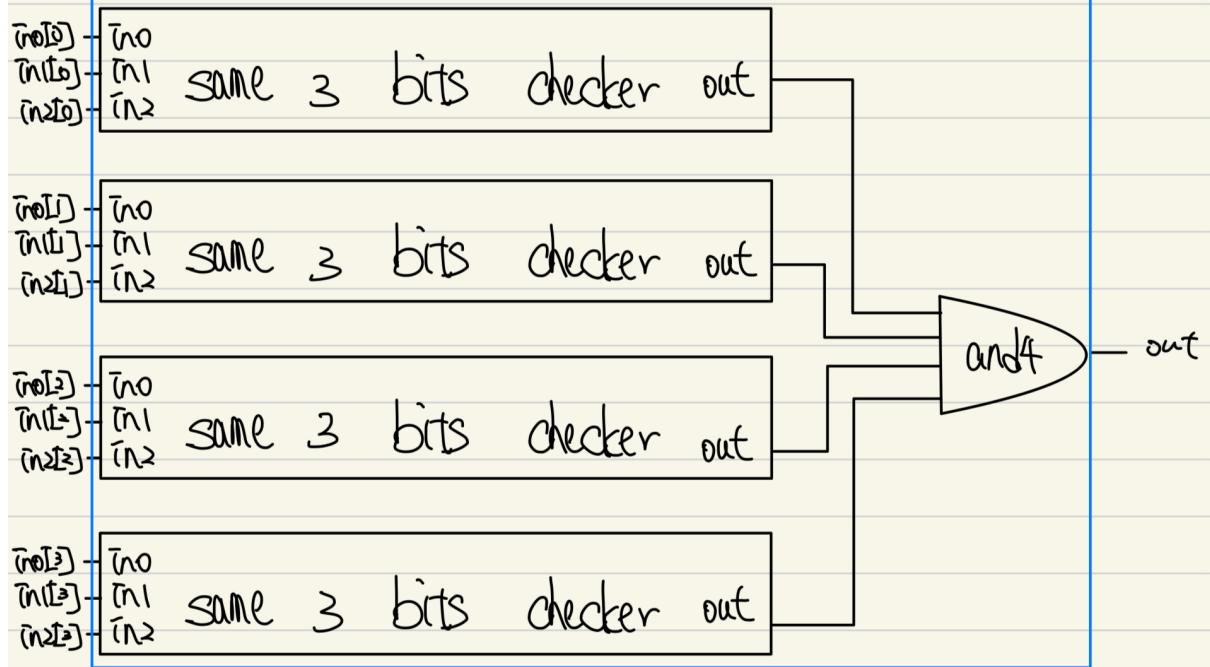


2 Pairs Possible Checker

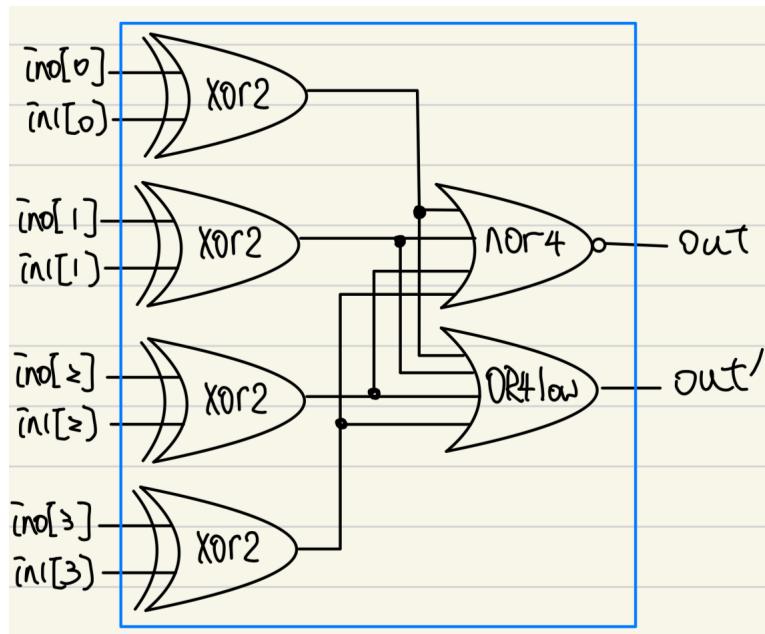


<p>2 Pairs Possible Sub Checker</p>	<p>$\bar{in}_0[3:0]$ $\bar{in}_0[3:0]$ identical 2 ranks checker $\bar{in}_1[3:0]$ $\bar{in}_1[3:0]$</p> <p>$\bar{in}_2[3:0]$ $\bar{in}_2[3:0]$ identical 2 ranks checker $\bar{in}_3[3:0]$ $\bar{in}_3[3:0]$</p> <p>$\bar{in}_0[3:0]$ $\bar{in}_0[3:0]$ identical 2 ranks checker $\bar{in}_1[3:0]$ $\bar{in}_1[3:0]$</p> <p>$\bar{in}_2[3:0]$ $\bar{in}_2[3:0]$ identical 2 ranks checker $\bar{in}_3[3:0]$ $\bar{in}_3[3:0]$</p> <p>$\bar{in}_0[3:0]$ $\bar{in}_0[3:0]$ identical 2 ranks checker $\bar{in}_1[3:0]$ $\bar{in}_1[3:0]$</p>
<p>1 Pair Possible Checker</p>	<p>$\bar{in}_0[3:0]$ $\bar{in}_0[3:0]$ identical 2 ranks checker $\bar{in}_1[3:0]$ $\bar{in}_1[3:0]$</p> <p>$\bar{in}_0[3:0]$ $\bar{in}_0[3:0]$ identical 2 ranks checker $\bar{in}_2[3:0]$ $\bar{in}_2[3:0]$</p> <p>$\vdots \times 10$</p> <p>$\bar{in}_3[3:0]$ $\bar{in}_3[3:0]$ identical 2 ranks checker $\bar{in}_4[3:0]$ $\bar{in}_4[3:0]$</p>
<p>Identical 4 Ranks Checker</p>	<p>$\bar{in}_0[0]$ $\bar{in}_0[0]$ same 4 bits checker out $\bar{in}_1[0]$ $\bar{in}_1[0]$ $\bar{in}_2[0]$ $\bar{in}_2[0]$ $\bar{in}_3[0]$ $\bar{in}_3[0]$</p> <p>$\bar{in}_0[1]$ $\bar{in}_0[1]$ same 4 bits checker out $\bar{in}_1[1]$ $\bar{in}_1[1]$ $\bar{in}_2[1]$ $\bar{in}_2[1]$ $\bar{in}_3[1]$ $\bar{in}_3[1]$</p> <p>$\bar{in}_0[2]$ $\bar{in}_0[2]$ same 4 bits checker out $\bar{in}_1[2]$ $\bar{in}_1[2]$ $\bar{in}_2[2]$ $\bar{in}_2[2]$ $\bar{in}_3[2]$ $\bar{in}_3[2]$</p> <p>$\bar{in}_0[3]$ $\bar{in}_0[3]$ same 4 bits checker out $\bar{in}_1[3]$ $\bar{in}_1[3]$ $\bar{in}_2[3]$ $\bar{in}_2[3]$ $\bar{in}_3[3]$ $\bar{in}_3[3]$</p>

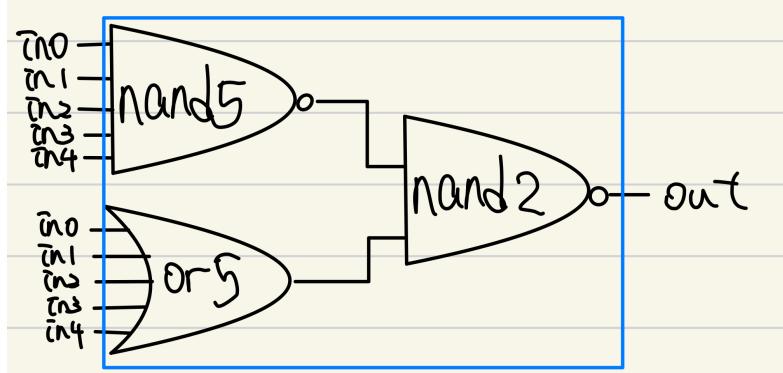
Identical
3 Ranks
Checker



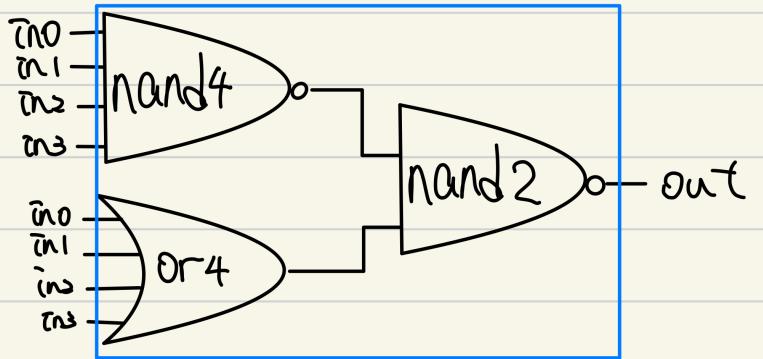
Identical
2 Ranks
Checker



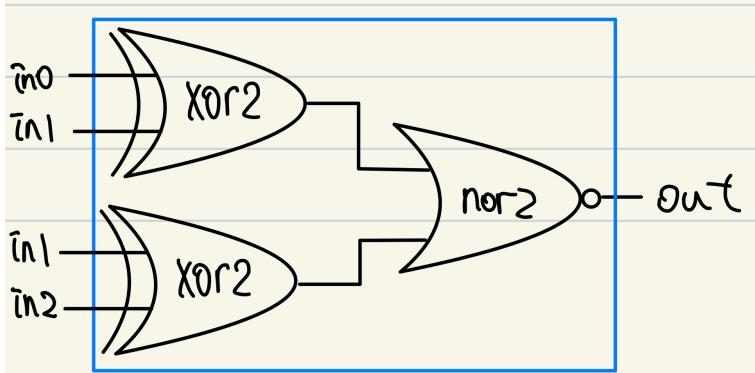
Same 5
Bits
Checker



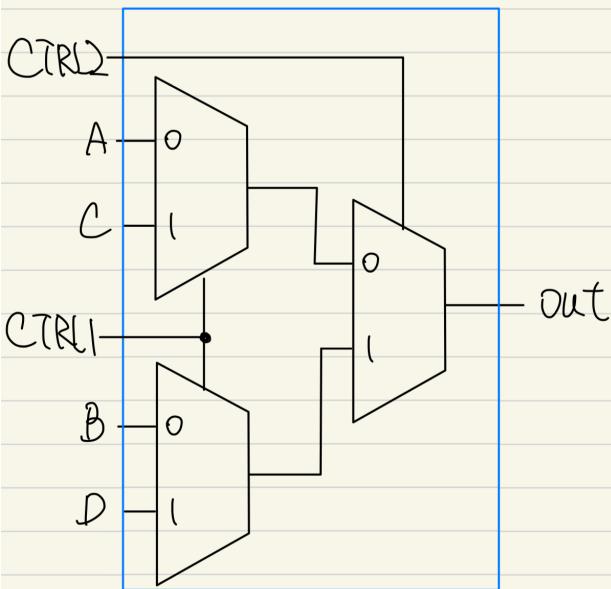
Same 4 Bits Checker



Same 3 Bits Checker



MUX4C



5. Self-defined Basic Gates

Name	Boolean Expression with the Lowest delay	Delay (ns)
XNOR2	$z = (a \wedge b)'$	$EO + IV = 0.470$
OR3low	$z = [(a+b)'c]'$	$NR2 + ND2 = 0.403$
OR4low	$z = [(a+b)'(c+c)]'$	$NR2 + ND2 = 0.403$
OR5	$z = [(a+b)'(c+d)'e]'$	$NR2 + ND3 = 0.453$
NR5	$z = (a+b)'(c+d)'e$	$NR2 + ND3 = 0.502$
AN5	$z = [(ab)' + (cd)' + e]'$	$ND2 + NR3 = 0.521$
ND5	$z = [(abc)(de)]'$	$AD3 + ND2 = 0.451$
OR6	$z = [(a+b)'(c+d)'(e+f)]'$	$NR2 + ND3 = 0.453$
NR6	$z = (a+b)'(c+d)'(e+f)'$	$NR2 + ND3 = 0.502$

AN6	$z = [(ab)' + (cd)' + (ef)']'$	$ND2 + NR3 = 0.521$
ND6	$z = [(abc)(def)]'$	$AD3 + ND2 = 0.451$
OR8	$z = (a+b+c+d)'(e+f+g+h)'$	$NR4 + AN2 = 0.570$
NR8	$z = [(a+b+c+d)'(e+f+g+h)']'$	$NR4 + ND2 = 0.521$
AN8	$z = [(abcd)' + (efgh)']'$	$AN4 + NR2 = 0.523$
ND8	$z = [(abc)(def)(gh)]'$	$AN3 + ND3 = 0.501$
AN9	$z = (abc)(def)(ghi)$	$AN3 + AN3 = 0.550$
ND9	$z = [(abc)(def)(ghi)]'$	$AN3 + ND3 = 0.501$
OR10	$z = [(a+b+c+0)'(d+e+f+0)'(g+h+i+j)']'$	$NR4 + ND3 = 0.571$
NR10	$z = (a+b+c+0)'(d+e+f+0)'(g+h+i+j)'$	$NR4 + AN3 = 0.620$
AN10	$z = [(abc)' + (def)' + (gh)' + (ij)']'$	$ND3 + NR4 = 0.571$
ND10	$z = [(abc)(def)(gh)(ij)]'$	$AN3 + ND4 = 0.571$

6. Discussion