

2015170063 오원석 논리설계 termproject1

Missionaries and cannibals problem

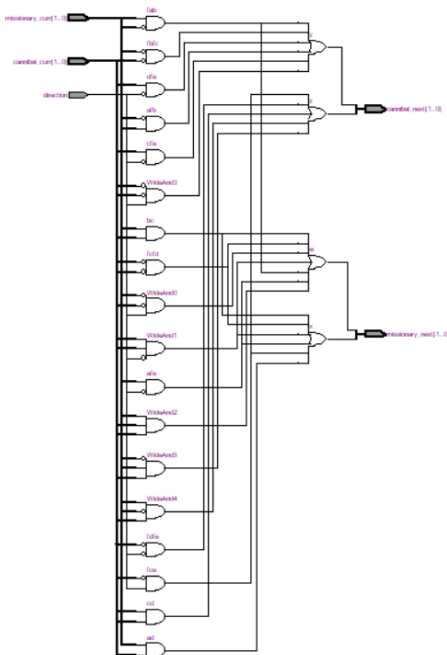
### Project summary

이 프로젝트는 3명의 사람과 3명의 식인종이 강을 건너는 문제이다. 강을 건너기 전의 상황을 input으로 두고 강을 건넌 뒤의 상황을 output으로 둔다. 그러면 각각의 종족들이 3명이기 때문에 2bit으로 표현이 가능하다. 따라서 움직이는 배를 input으로 포함을 시켜서 5개의 input(2개씩 묶이기 때문에 실질적으로는 3개) 그리고 4개의 output(2개씩 묶이기 때문에 실질적으로는 2개)이 나오게 된다. 될 수 있는 모든 상황을 파악하고 이를 시뮬레이션 하고 결과를 도출해내는 것이 이번 프로젝트의 목표이다.

### Module description

이번 모듈은 missionary\_curr, cannibal\_curr, direction의 input 그리고 missionary\_next, cannibal\_next의 output을 구하는 과정을 진행하면 된다. 중간에 andgate 그리고 두개의 orgate를 거쳐서 최종 output 두개로 나온다는 사실을 알 수 있다.

다음은 Quartus로 logicdiagram을 그리게 되면 왼쪽과 같이 나오게 된다.



Result

Simulation result

결과는 다음과 같다.

input					output				order
m1	m0	c1	c0	d	mn1	mn0	cn1	cn0	
0	0	0	0	0	1	1	1	1	
0	0	0	0	1	1	1	1	1	
0	0	0	1	0	0	0	1	0	10
0	0	0	1	1	1	1	1	1	
0	0	1	0	0	0	0	1	1	8
0	0	1	0	1	0	0	0	0	11
0	0	1	1	0	1	1	1	1	
0	0	1	1	1	0	0	0	1	9
0	1	0	0	0	1	1	1	1	
0	1	0	0	1	1	1	1	1	
0	1	0	1	0	1	0	1	0	6
0	1	0	1	1	1	1	1	1	
0	1	1	0	0	1	1	1	1	
0	1	1	0	1	1	1	1	1	
0	1	1	1	0	1	1	1	1	
0	1	1	1	1	1	1	1	1	
1	0	0	0	0	1	1	1	1	
1	0	0	0	1	1	1	1	1	
1	0	0	1	0	1	1	1	1	
1	0	0	1	1	1	1	1	1	
1	0	1	0	0	1	1	1	1	
1	0	1	0	1	0	0	1	0	7
1	0	1	1	0	1	1	1	1	
1	0	1	1	1	1	1	1	1	
1	1	0	0	0	1	1	0	1	4
1	1	0	0	1	1	1	1	1	
1	1	0	1	0	1	1	1	0	2
1	1	0	1	1	0	1	0	1	5
1	1	1	0	0	1	1	1	1	

1	1	1	0	1	1	1	0	0	3
1	1	1	1	0	1	1	1	1	
1	1	1	1	1	1	1	0	1	1

m1= missionary\_curr[1]  
 m0= missionary\_curr[0]  
 c1= canibal\_curr[1]  
 c0= canibal\_curr[0]

mn1= missionary\_next[1]  
 mn0= missionary\_next[0]  
 cn1= canibal\_next[1]  
 cn0= canibal\_next[0]

### Synthesis Availability

위의 결과를 통해 output이 1111이 나오는 경우는 있을 수 없거나 돌아가는 경우 즉, present state is far from the solution인 경우이다. 이때는 다시 처음으로 돌아가게 되는 것을 알 수 있다. 나머지의 경우들을 살펴보면

Order	Left	Boat	Right	Input	output
0	MMM CCC				
1	MMMC	CC ->	CC	1 1 1 1 1	1 1 0 1
2	MMMCC	<- C	C	1 1 0 1 0	1 1 1 0
3	MMM	CC ->	CCC	1 1 1 0 1	1 1 0 0
4	MMMC	<- C	CC	1 1 0 0 0	1 1 0 1
5	MC	MM ->	MM CC	1 1 0 1 1	0 1 0 1
6	MMCC	<- MC	M C	0 1 0 1 0	1 0 1 0
7	CC	MM ->	MMM C	1 0 1 0 1	0 0 1 0
8	CCC	<- C	MMM	0 0 1 0 0	0 0 1 1
9	C	CC ->	MMM CC	0 0 1 1 1	0 0 0 1

10	CC	<- C	MMM C	0 0 0 1 0	0 0 1 0
11		CC ->	MMM CCC	0 0 1 0 1	0 0 0 0

이로써 옆의 상황과 11가지의 경우에 같게 시뮬레이션 결과가 나오는 것을 확인할 수 있다.