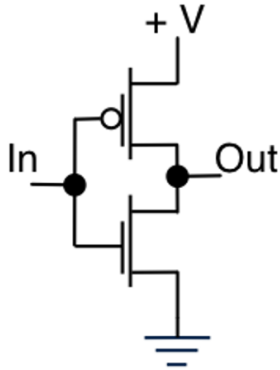


1.
10점
(+1점)

(1) NOT gate(총 5점)



올바른 구현 시 5점.

여기서 올바른 구현이란, input signal을 넣었을 때 output signal이 gate의 진리표에 따라 작동하는 구현을 의미한다.

nMOS, pMOS의 표현이 잘못되었을 시 -3점.

nMOS, pMOS의 표현은 위의 표현만을 올바른 표현으로 인정한다.

input/output signal의 “표현”이 불분명/잘못되었을 시 -1점.

signal이 들어가고 나오는 위치가 올바르다면 signal의 이름은 어떤 것이든 상관없다. 그러나 만약 signal이 들어가고 나오는 위치가 올바르지 않고 회로가 정상적인 output을 내지 않는다면 감점 규칙의 적용 없이 0점 처리한다.

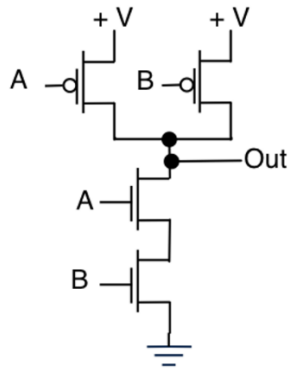
vcc와 ground “표현”이 불분명/잘못되었을 시 -1점.

만약 vcc와 ground의 연결이 잘못되었다면 회로가 정상적으로 작동하지 않으므로 감점 규칙을 적용하지 않고 0점 처리한다.

최소 0점.

위의 잘못된 부분만을 제대로 수정하였을 때 올바른 구현이 되는 경우에만 위의 감점 규칙을 적용한다. 그렇지 않으면 감점 규칙을 적용하지 않고 0점 처리한다.

(2) NAND gate(총 3점)



올바른 구현 시 3점.

여기서 올바른 구현이란, input signal을 넣었을 때 output signal이 gate의 진리표에 따라 작동하는 구현을 의미한다.

nMOS, pMOS의 표현이 잘못되었을 시 -1점.

nMOS, pMOS의 표현은 위의 표현만을 올바른 표현으로 인정한다.

input/output signal이 불분명/잘못되었을 시 -1점.

signal이 들어가고 나오는 위치가 올바르다면 signal의 이름은 어떤 것이든 상관없다. 그러나 만약 signal이 들어가고 나오는 위치가 올바르지 않고 회로가 정상적인 output을 내지 않는다면 감점 규칙의 적용 없이 0점 처리한다.

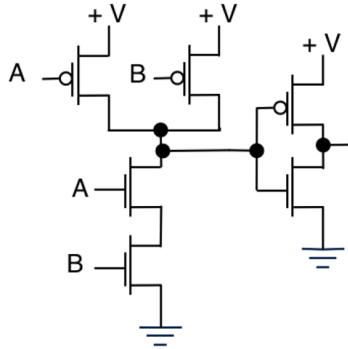
vcc와 ground “표현”이 불분명/잘못되었을 시 -1점.

만약 vcc와 ground의 연결이 잘못되었다면 회로가 정상적으로 작동하지 않으므로 감점 규칙을 적용하지 않고 0점 처리한다.

최소 0점.

위의 잘못된 부분만을 제대로 수정하였을 때 올바른 구현이 되는 경우에만 위의 감점 규칙을 적용한다. 그렇지 않으면 감점 규칙을 적용하지 않고 0점 처리한다.

(3) AND gate(총 2점)



올바른 구현 시 2점.

여기서 올바른 구현이란, input signal을 넣었을 때 output signal이 gate의 진리표에 따라 작동하는 구현을 의미한다

nMOS, pMOS의 표현이 잘못되었을 시 -1점.

nMOS, pMOS의 표현은 위의 표현만을 올바른 표현으로 인정한다.

input/output signal이 불분명/잘못되었을 시 -1점.

signal이 들어가고 나오는 위치가 올바르다면 signal의 이름은 어떤 것이든 상관없다. 그러나 만약 signal이 들어가고 나오는 위치가 올바르지 않고 회로가 정상적인 output을 내지 않는다면 감점 규칙의 적용 없이 0점 처리한다.

vcc와 ground 표현이 불분명/잘못되었을 시 -1점.

만약 vcc와 ground의 연결이 잘못되었다면 회로가 정상적으로 작동하지 않으므로 감점 규칙을 적용하지 않고 0점 처리한다.

composition이라는 keyword가 언급되면 보너스 점수 1점.

최소 0점.

위의 잘못된 부분만을 제대로 수정하였을 때 올바른 구현이 되는 경우에만 위의 감점 규칙을 적용한다. 그렇지 않으면 감점 규칙을 적용하지 않고 0점 처리한다.

nMOS, pMOS 트랜지스터를 이용하지 않은 구현은 소문항에 관계없이 모두 0점 처리한다. (예: logic gate만을 명시한 경우)

1번 문항에서 얻을 수 있는 보너스 점수는 최대 1점이다.

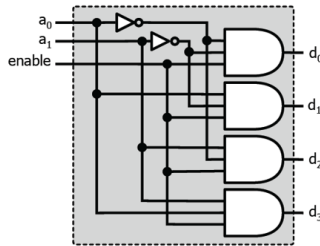
총 11점(보너스 점수 1점 포함).

2.

10점
(+2점)

진리표와 External interface

enable	a ₁	a ₀	d ₃	d ₂	d ₁	d ₀
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0
0	X	X	0	0	0	0



(1) 진리표를 올바르게 기술하였을 시 3점.

진리표 내에서 input signal 간의 순서와 output signal 간의 순서는 2-to-4 decoder의 semantic만 통한다면 진리표는 올바른 것으로 본다.

단, 이후의 소문항들에서 학생이 작성한 진리표에 근거하여 올바르게 연결되어 있는지에 대한 채점 기준으로 고려한다.

수정사항(2015-05-04 18:40)

Enable signal만 없다면 -1점, 기타 minor error -1점

(2) External Interface를 올바르게 기술하였을 시 2점.

Input signal들을 올바르게 기술하였다면 1점.

enable, a₀, a₁, 또는 그에 준하는 세 input signal이 모두 명시되어 있어야 점수가 주어진다.

Output signal들을 올바르게 기술하였다면 1점.

d₀, d₁, d₂, d₃, 또는 그에 준하는 네 output signal이 모두 명시되어 있어야 점수가 주어진다.

진리표와는 별도로 각 signal들의 역할이 올바르게 설명되어 있다면 **보너스 점수 1점**이 주어진다. 역할의 설명이 올바르다면 이름은 어떤 것이 되어도 상관없으며, 내부 구조의 논리 게이트의 올바른 명시만으로도 각 signal의 역할 설명은 충분하다.

내부 구조의 논리 게이트가 명시되어 있지 않을 경우 다음과 같은 설명이 상세히 명시되어 있다면 **보너스 점수 1점**이 부여된다.

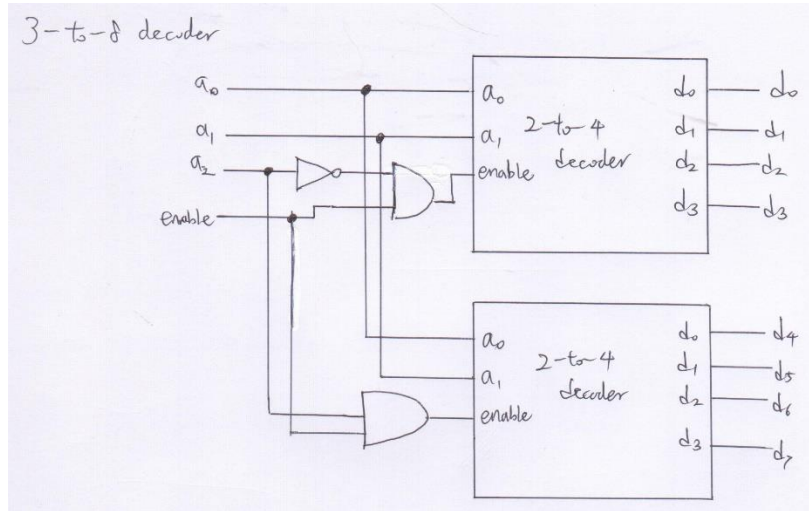
Signal들의 역할 설명 예시 답안

“위의 그림에서 a₀와 a₁, 또는 그에 준하는 input signal들은 d₀~d₃, 또는 그에 준하는 output signal들 중 한 output만 1이 되도록 선택하는 역할을 담당한다.

Enable signal, 또는 그에 준하는 signal이 1일 때에만 위의 선택이 이루어지며, enable signal이 0일 때에는 다른 input signal의 값에 상관없이 모든 output signal의 값이 0이 된다.”

이 소문항에서 최대로 얻을 수 있는 보너스 점수는 1점이다.

(3) 2-to-4 decoder to 3-to-8 decoder(3점)



올바른 구현시 3점.

여기서 올바른 구현은, 올바른 진리표에 따라 input signal 값을 넣으면 output signal 값이 올바른 진리표에 따라 나오는 구현을 의미한다.

Enable signal이 구현되어 있지 않을 시 -1점.

Enable signal이 3-to-8 decoder 바깥쪽에서도 들어와야 감점이 되지 않는다.

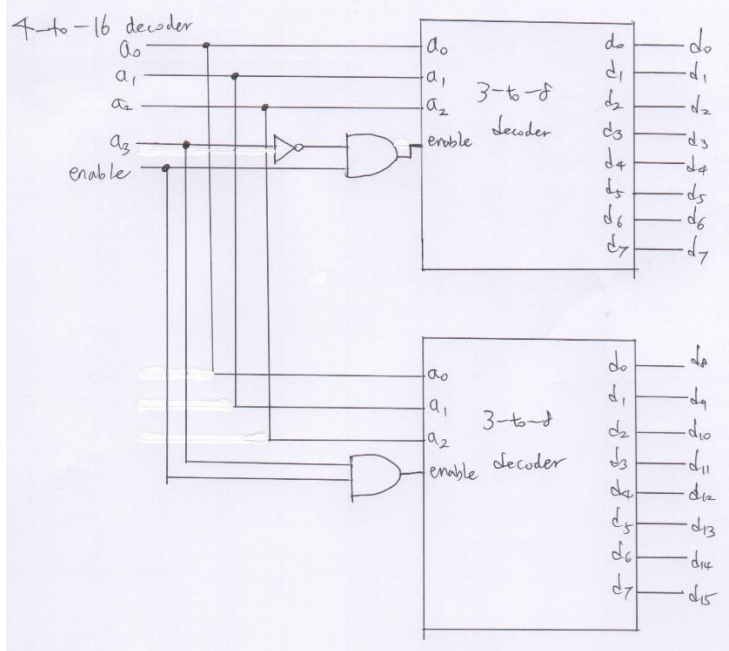
구현의 올바른 의미는 통하지만 위에 제시된 내용 외의 기타 minor error 가 있다면 -1점.

수정사항 (2015-05-06 11:23 수정)

input/output signal이 불분명/잘못되었을 시 -1점.

signal이 들어가고 나오는 위치가 올바르다면 signal의 이름은 어떤 것이든 상관없다. 그러나 만약 signal이 들어가고 나오는 위치가 올바르지 않고 회로가 정상적인 output 을 내지 않는다면 감점 규칙의 적용 없이 0점 처리한다.

(4) 3-to-8 decoder to 4-to-16 decoder(2점)



올바른 구현시 2점.

여기서 올바른 구현은, 올바른 진리표에 따라 input signal 값을 넣으면 output signal 값이 올바른 진리표에 따라 나오는 구현을 의미한다.

Enable signal이 구현되어 있지 않을 시 -1점.

Enable signal이 4-to-16 decoder 바깥쪽에서도 들어와야 감점이 되지 않는다.

구현의 올바른 의미는 통하지만 minor error 가 있다면 -0.5점.

수정사항 (2015-05-06 11:23 수정)

input/output signal이 불분명/잘못되었을 시 -1점.

signal이 들어가고 나오는 위치가 올바르다면 signal의 이름은 어떤 것이든 상관없다. 그러나 만약 signal이 들어가고 나오는 위치가 올바르지 않고 회로가 정상적인 output을 내지 않는다면 감점 규칙의 적용 없이 0점 처리한다.

2번 문항 전체에 걸쳐서 build-upon 이라는 keyword가 한 번이라도 언급되면 보너스 점수 1점.

2번 문항에서 얻을 수 있는 보너스 점수는 최대 2점이다.

총 12점(보너스 점수 2점 포함).

3.

10점

(+1점)

8개의 n-bit register를 담고 있는 register file.

8개의 n-bit register 중에서[1점] 2개의 register의 값을 동시에 읽을 수 있으며[1점], 그와 동시에 1개의 register의 값을 쓸 수 있다.[1점]

(read는 clock에 무관하다는 사실을 언급하면 보너스 1점)

A_address와 B_address를 통해 읽을 register의 번호를 2개 지정하여[1점] 각각 A_data와 B_data를 통해 register의 값 2개를 동시에 읽어낼 수 있다[1점]. 한편, write signal을 enable하고[1점] write할 register의 번호를 W_address를 통해 주면서[1점], 동시에 W_data를 통해 write할 data를 주면[1점], clock의 rising edge 때에[1점] W_address로 지정한 번호의 register의 값이 W_data의 값(위에서 준 data 값)으로 바뀌게 된다.[1점]

각 부분이 서술에 포함되어 있을 경우에 각 부분에 대해 점수를 부여한다.

3번 문항에서 받을 수 있는 보너스 점수는 최대 1점이다.

수정사항(2015-05-04 18:40)

8개의 n-bit register는 "8개의", 혹은 "n-bit register"가, 혹은 그에 준하는 의미가 언급되면 점수 인정

총 11점(보너스 점수 1점 포함).

4.
10점

	AA	BA	DA	AS	BS	FS	DS	RW	MW
(1)	001	XXX	010	1	X	00000	0	1	0
(2)	001	010	000	1	0	00010	0	1	0
(3)	001	XXX	000	1	X	XXXXX	1	1	0

(1) R2 <- R1 (3점)

올바르게 값들을 기술했으면 3점

Don't care value에 대해서는 어떤 값을 써 넣어도 감점 없음.

위의 component 기준에서 한 component 틀릴 때마다 -1점.

최소 0점.

(2) R0 <- R1 + R2 (3점)

올바르게 값들을 기술했으면 3점

Don't care value에 대해서는 어떤 값을 써 넣어도 감점 없음.

위의 component 기준에서 한 component 틀릴 때마다 -1점.

최소 0점.

(3) R0 <- M[R1] (4점)

올바르게 값들을 기술했으면 4점

Don't care value에 대해서는 어떤 값을 써 넣어도 감점 없음.

	<p>위의 component 기준에서 한 component 틀릴 때마다 -1점. 최소 0점.</p> <p>총 10점.</p>
--	--

1번~4번 총 40점(보너스 점수 4점 별도)

2015-1학기 컴퓨터의 개념 및 실습 중간시험 채점 기준표	
5. 10점	<p>Bubble sort 알고리즘의 Python 기술 (총 6점)</p> <p>시간복잡도 n^2을 따르며, 서로 이웃한 원소끼리 비교해 뒤로 보내며 정렬하는 방식으로 알고리즘을 작성했다. (4점)</p> <p>Python에서 Syntax Error없이 실행된다. (2점)</p> <p>[-2,1,5,2,7] 에 대한 Bubble sort 동작 설명 (총 4점)</p> <p>서로 이웃한 원소끼리 비교해 뒤로 보내는 방식을 설명했다. (2점)</p> <p>서로 이웃한 원소끼리 비교해 뒤로 보내는 방식이 반복적으로 일어나는 것을 제대로 설명했다. (2점)</p>

6-10. 50점	<p>(각 5점)</p> <p>(1) assignment statement assignment 가 들어가면 4점, assignment없으면 0점</p> <p>(2) expression</p> <p>(3) variables</p> <p>(4) operators operations, operands - 0점</p> <p>(5) recursion</p> <p>(6) return fibonacci(n-1) + fibonacci(n-2) 의미는 맞지만 syntax 틀린 경우 1점</p> <p>(7) method</p> <p>(8) x>0</p> <p>(9) nested list nesting list, nest list - 1점</p> <p>(10) return</p>
--------------	--

5번~10번 총 60점

수정사항 (2015-05-08 11:25 수정)

부분점수 채점 기준표 반영

총 100점 만점(보너스 점수 4점 별도)