underflow

float 같은 경우 정보를 저장할 떄

0.38\*10^(-20)

|  |  |
| --- | --- |
| 소수부분:0.38 | 지수:-20 |

오버플로우 무한대 값

언더플로우 0에 가까운 값

오버플로우 덧셈의 절댓값이 해당 범위를 넘어서면 표현 불가능

언더 플루우: 실수(float)의 지수 부분의 비트 수가 제한되기 때문에 음의 절댓값이 주어진 비트보다 클 때 발생한다.

ex) 0.38\*10^(-20)을 소수 지수 부분이 각각 4비트인 저장공간에 저장한다고 하면

0.38은 소수니깐 ok, but 지수 부분은 |-20| = 20 근데 이 값은 4비트를 벗어나므로 저장 X

Magnitue comparator

0011

0101 둘 중에 뭐가 큰지 어떻게 아는가? 그냥 앞에서부터 순서대로 비교

(a'b+ab')' = (a'+b)(a+b')=a'b'+ab

중요 디코더!

n개의 입력 2^n개의 민텀

bcd adder , : 0~9 + 0~9 = 0~19

4 bit: binary adder 차이점: 0~15 + 0~15 = 0~31

encoder: decoder의 반대 2^n개를 넣으며 n개가 나옴

priority encoder: 생략(강의 노트에서 삭제했음)

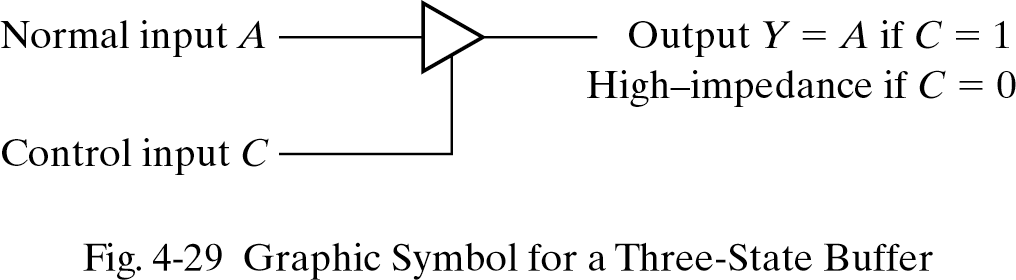
Multiplexers: MUX라고도 함 입력인 2^n개가 들어왔을 때 한놈만 내보내는 것

MUX를 써서 설계하는 방법(기말)

왼쪽에 truth table 오른쪽에 MUX

MUX 그림은 4\*1 => 입력이 4개 출력이 1개 입력이 4개이면 select가 2개 8개면 select가 3개.....

truth table을 2개씩 나누어준다 F를 만들어준다



c가 1이면 정상적으로 작동

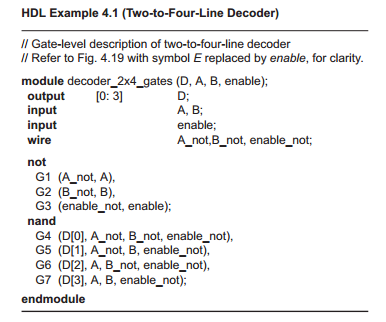
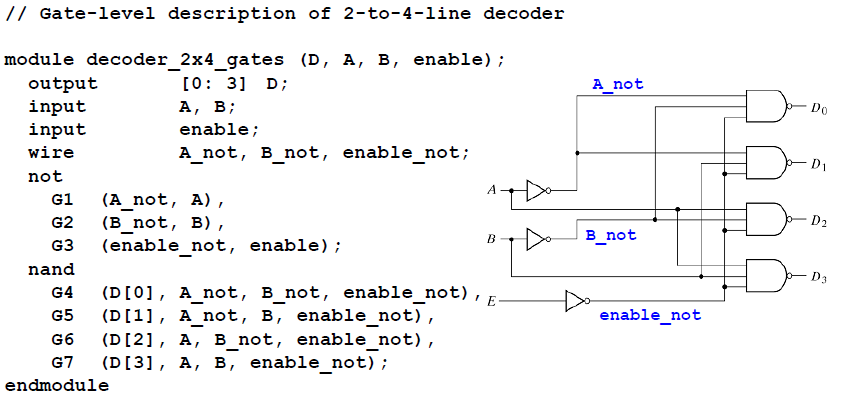
c가 0이면 open circuit이라고 하며 이때는 값이 정상적으로 전달이 안됨

HDL

assign 중요 이걸로 그릴줄 알아야 함

always

HDL의 세가지 기법을 설명하시오: gate level modeling , data modeling: assign, behavioral modeling: always



D가 4개니깐 벡터처럼 [0:3]으로 표현함

\*full adder가 설계되었다는 가정 하에 4-bit adder을 설계하라 (기말고사 문제로 잘냄)

{ }: concatenation

?: : conditional

4-bit adder