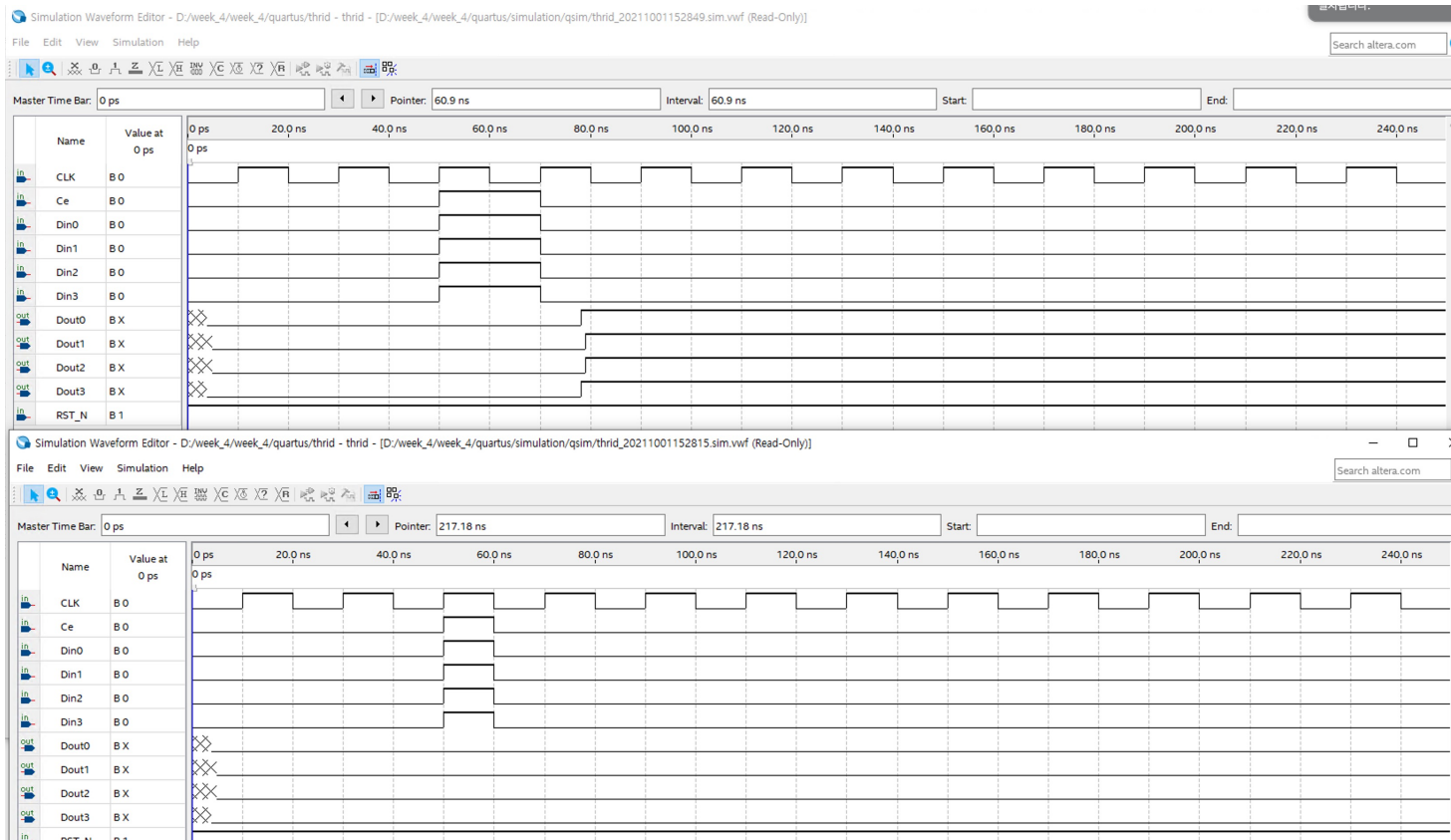


4주차 문제

우리가 Input을 준다면 그 때 선 (wire 면서 Net)이라고 불리는 곳을 지냅니다. 이때도 delay가 실제로 존재하는데 Functional Simulation에서는 이것을 생각하지 않기 때문에, Positive edge일때 어떻게 동작할지 몰라서 랜덤적으로 뽑혀서 설명한 대로 동작하지 않고, 마음대로 1~2개씩 잘못된 값을 보여준거라고 생각하세요. Timing simulation을 하면 이것들을 고려해서 제가 앞서 설명 드린 대로 clk의 posedge edge와 같이 ce와 data를 줄 시에는 아래와 같이 동작하지 않는게 정상입니다.



이런 현상을 보고 싶지 않으신분은 수업 때 이야기 한 대로 posedge 이전에 Ce를 1로 만들면 functional simulation을 써도 무방합니다.

실습 1

10진 카운터 2개를 사용하여 100진 카운터를 설계하시오

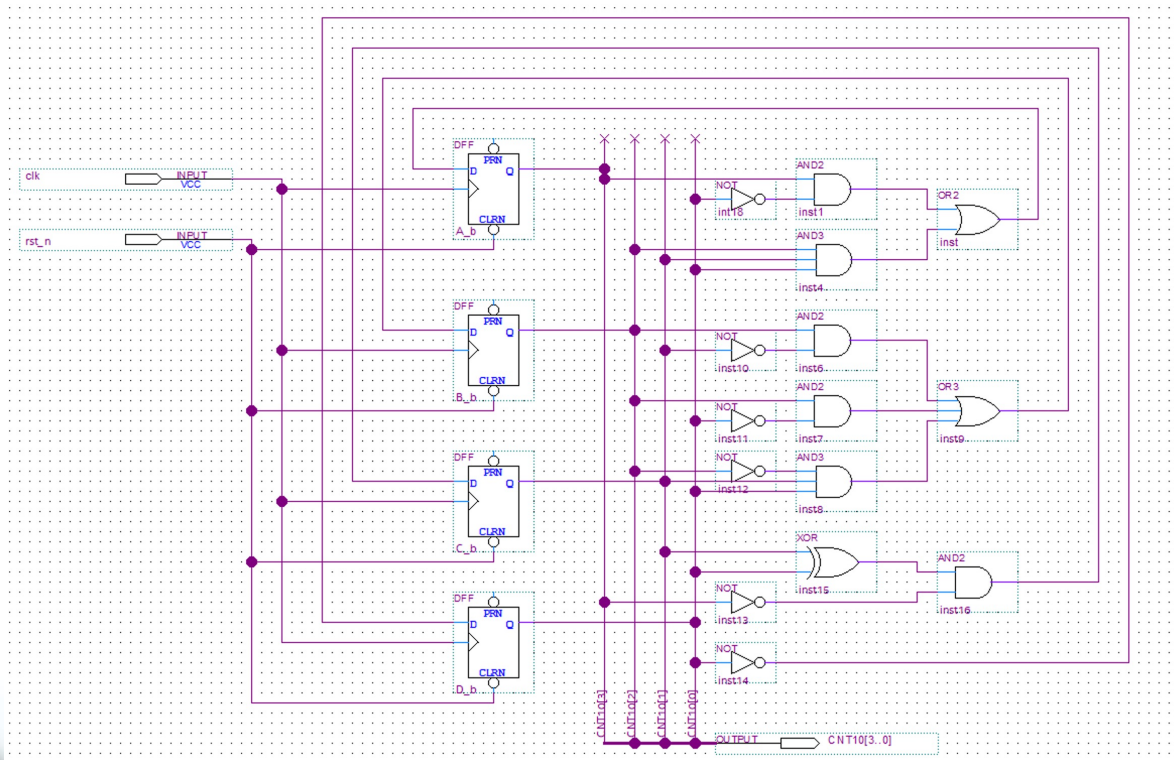
입력 : clk, rst

출력 : [7:0] CNT100

= {bf_CNT100[7:4], bf_CNT10[3:0]}

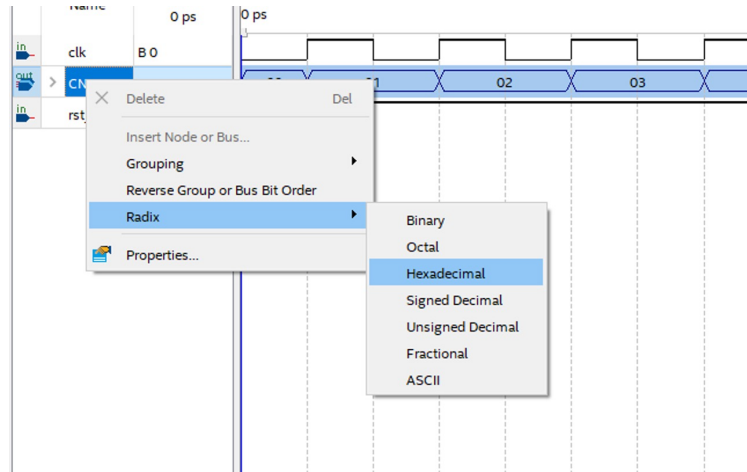
CNT10은 0~9까지 값을 의미하도록 출력

CNT100은 10의 배수 값을 의미하도록 출력

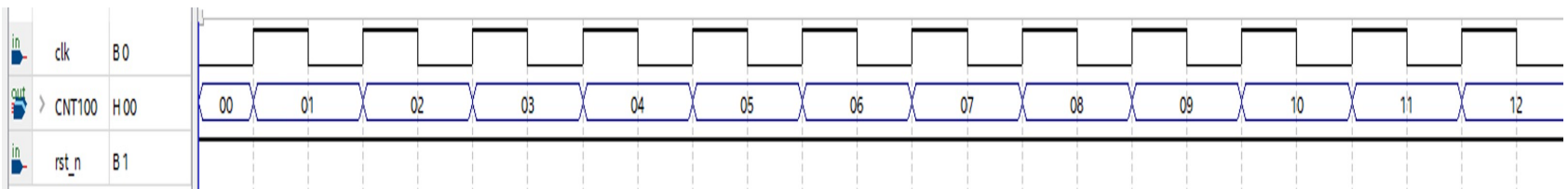


10진 카운터의 회로도

실습 1

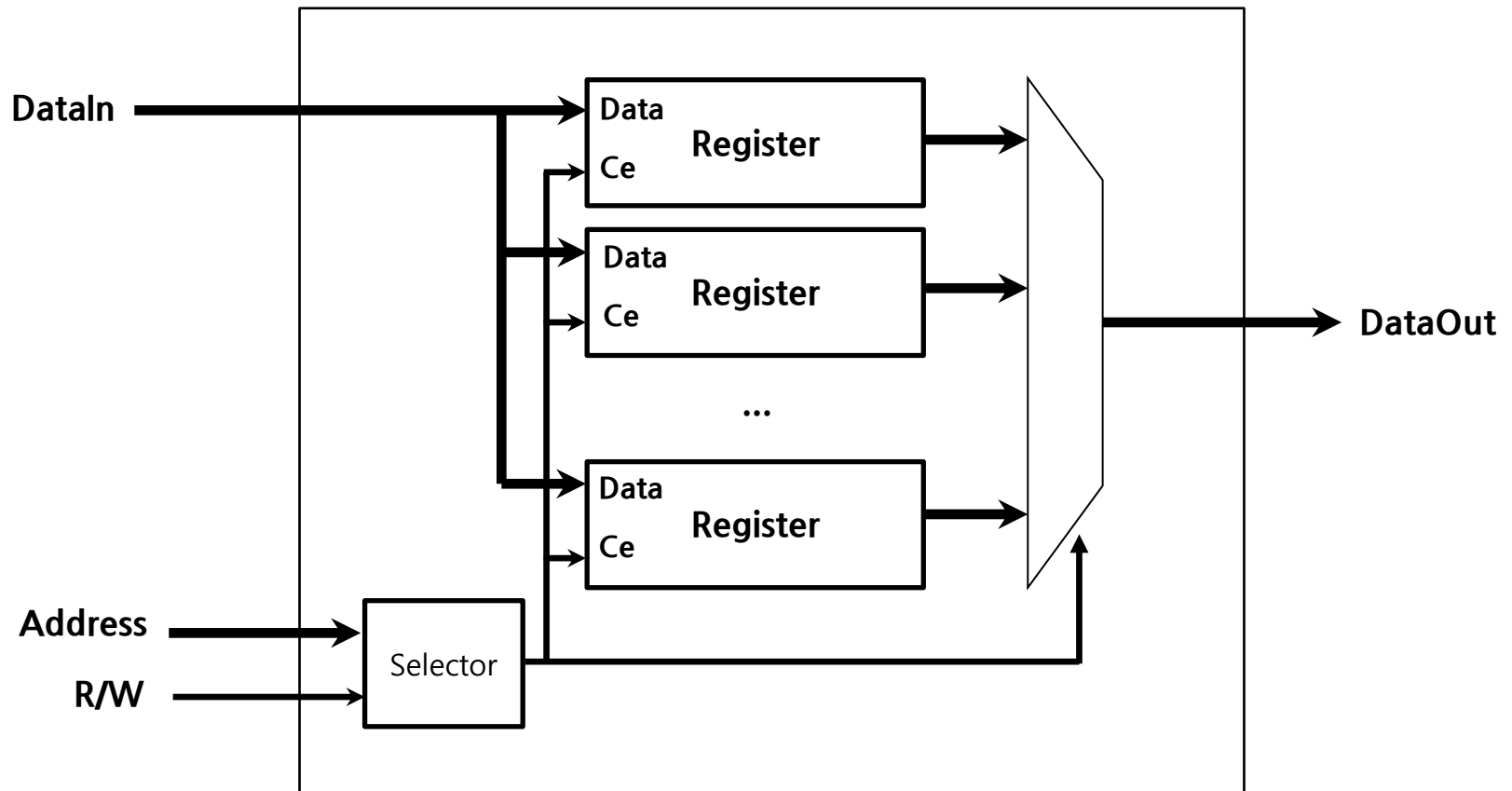


Hexadecimal로 값을 봐야함
bit로 99 까지 counter 하는 개념이 아님 !



실습 2

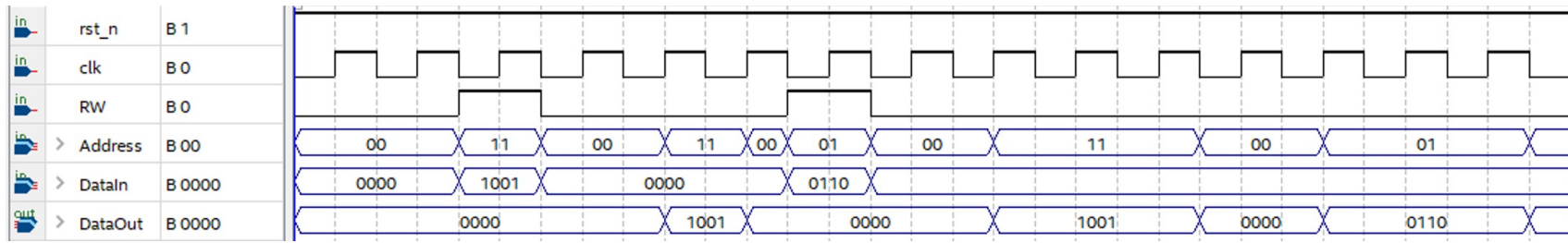
4bit Register 4개를 사용하여 4bit x 4 SRAM을 설계하시오



SRAM의 블록 다이어그램

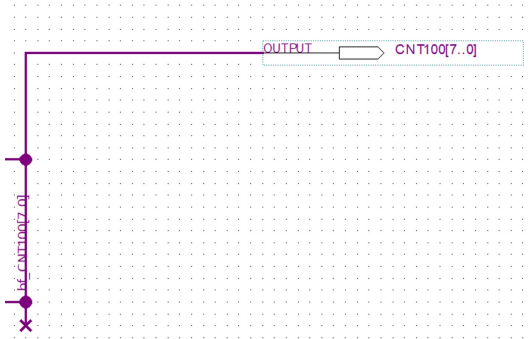
실습 2

시뮬레이션 결과

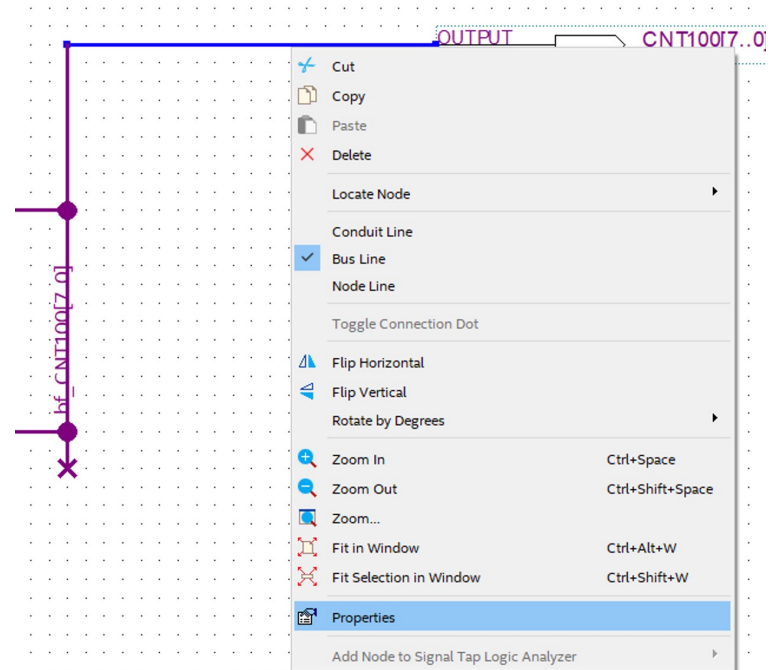


참고사항

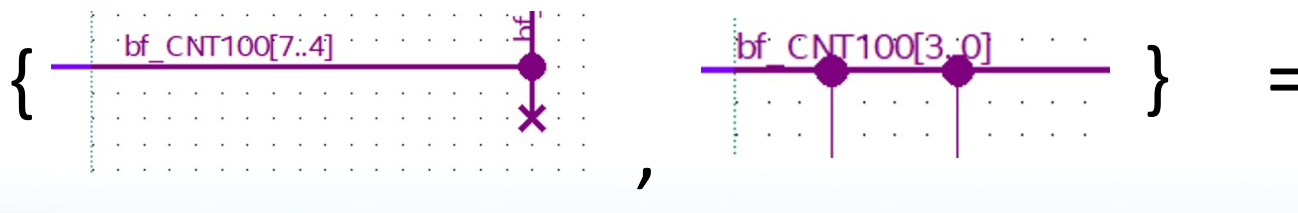
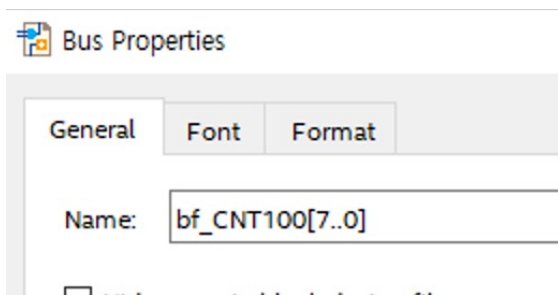
Bus node 사용시 다중 bit 입출력 가능



Wire 우클릭 후 properties 선택

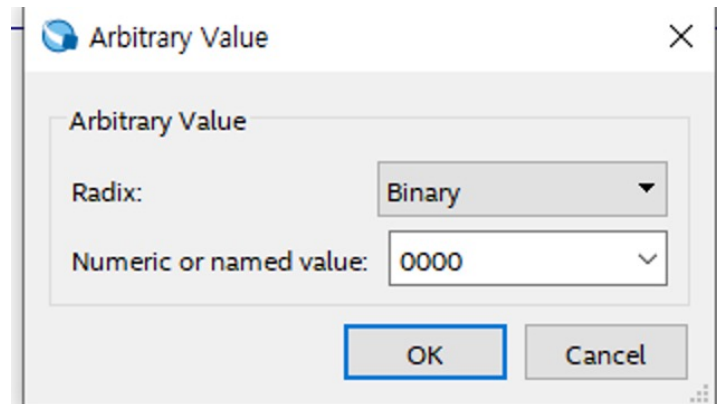


이런식으로 선언시 8bit wire 가 됨





Arbitrary Value를 사용하면 다중 bit 입력을 줄 수 있음



Binary로 4bit 입력을 줄 수 있는 것

다중 입출력 I/O로 생성시 Symbol에서도 편리함

