

1. Processor가 Instruction을 실행하는 과정에서 datapath와 control의 역할 분담에 대하여 설명하되, 'add' instruction을 인용하여야 한다.
2. 슬라이드 4-19의 datapath를 출력(복사)하여 다음 instruction에서 의미 있는 data가 통과하는 경로와 element를 표시하시오. 가능하면 검정색 이외의 색을 사용하여 원래의 datapath와 구분하시오.
 - (a) add
 - (b) sw
 - (c) beq
3. 교과서의 Main control + ALU control 대신 단일 Control 만으로도 동일한 기능을 구현할 수 있다. 다음 물음에 답하시오.
 - (a) 입력과 출력을 결정하시오. 슬라이드 4-23을 참조하여 비트 범위까지 정확하게 표현하시오.
 - (b) 슬라이드 4-21과 슬라이드 4-29를 참조하여 단일 Control의 동작 방식을 진리표로 작성하시오. Instruction은 슬라이드 4-29처럼 symbol로 표시할 수 있다.
 - (c) 조합회로의 크기를 복잡도를 예측하는 방법중의 하나는 비트 단위로 계산된 진리표의 크기를 계산하는 것이다. 예를 들면 입력이 A 비트이고 출력이 B 비트이면 진리표의 크기는 $2^A \times B$ 가 된다. 교과서에서의 방법과 단일 Control 방법의 크기를 비교하시오.
4. 교과서에 제시된 datapath와 control을 확장하여 addi instruction도 실행할 수 있도록 개선하려고 한다. 물음에 답하시오.
 - (a) 슬라이드 4-19의 datapath를 출력(복사)하여 추가할 부분을 보이시오. 가능하면 다른 색으로 구분하시오.
 - (b) 슬라이드 4-29의 진리표에서 추가할 부분을 보이시오.
5. 슬라이드 4-31에서의 datapath는 jump instruction도 실행할 수 있도록 확장된 것이다. 슬라이드 4-29의 진리표를 확장하여 jump instruction도 제어할 수 있도록 개선하시오.
6. 교과서에 제시된 datapath와 control을 단순화하려고 한다. 프로세서가 실행하는 명령어 집합에서 beq instruction과 sw instruction이 필요하지 않다고 가정한다.
 - (a) 슬라이드 4-19의 datapath를 참고하여 단순화된 datapath를 다시 그리시오.
 - (b) 슬라이드 4-29의 진리표에서 불필요한 부분을 제외하고 다시 쓰시오. 일부 control signal이 제거될 수 있음에 주의하시오.
7. "Stuck-at-one fault"는 신호가 항상 1의 값을 갖게 되는 결함을 가리킨다. 예를 들어서 RegWrite 신호에 stuck-at-one fault가 있다면 외부에서 주어진 값과 상관없이 RegWrite

신호에는 항상 1이 인가된 것과 같은 효과가 발생되므로 특정 명령어가 (모든 명령어가 아니라) 오동작하게 된다. 슬라이드 4-31에서의 datapath에 대하여 add, sub, and, or, lw, sw, beq, j 명령어를 대상으로, 아래 신호들에 대하여 stuck-at-one fault가 발생할 때 제대로 동작하지 않는 명령어들을 나열하고 이유를 설명하시오.

- (a) RegWrite
- (b) RegDst
- (c) ALUSrc
- (d) MemWrite
- (e) Branch