(page 1)

안녕하세요 저는 13주차 발표를 맡은 9조 20201635 전찬 이고,

(page 2)

shift register, ring counter, ,,,, 그리고 추가적인 이론에 대해 발표를 진행하겠습니다.

(page 3)

첫 번째로, shift register 는 데이터의 저장, 전송에서 중요한 역할을 하는 회로로, 여러 flip-flop의 조합으로 구성되게 됩니다. 아래는 가장 기본적인 shift register의 구조인데, 각 Clock 마다 저장된 data가 오른쪽 방향으로 이동하는 형태임을 파악할 수 있습니다. 이때 register 내에서 각 flip-flop은 stage라고 하기도 합니다.

(page 4)

이해를 위해 input에 0, 1, 0 ,1 을 순차적으로 대입했을 때 4bit shift register 의 동작을 설명해 보겠습니다. clock = 1일 때 Q0에는 0이 들어가며, 이는 화살표 방향으로 Q3에, clock = 2일 때 들어간 1은 화살표 방향으로 Q2에, clock = 3일 때 들어간 0은 화살표 방향으로 Q1에 저장되게 됩니다. 따라서 clock = 4일 때 모든 4bit data가 register 안에 저장됩니다.

(page 5)

이후에는 4~7번째 clock까지 Q3 output으로 0, 1, 0, 1이 순차적으로 나오는 것을 파악할 수 있습니다. 이렇게 0, 1, 0, 1이 몇 번의 clock 이후에 Q3을 통해서 나오기 때문에 shift register은 어떠한 data에 대해 의도적으로 시간을 지연시킬 때 활용하기도 합니다.

(page 6)

두 번째로, Ring counter 이란 shift register를 활용한 동기식 counter로, 마지막 output 이 다시 첫 번째 flip-flop의 input에 들어가는 고리 형태 구조입니다.

아래는 10 bit ring counter의 구조인데, 기본적으로는 preset, clear input 을 통해 Q0 ~ Q9을 1 0 0 0 0 0 0 형태를 만든 이후 counting을 시작하는 형태입니다.

(page 7)

아래 표는 clock pulse가 진행됨에 따라서 ring counter의 각 output을 표시한 것인데, clock pulse가 0일 때는 preset / clear 에 의해서 Q0 = 1인 형태임을 파악할 수 있습니다. 이후에는 shift register 형태를 따라 각 clock 마다 1이 오른쪽으로 이동하는 형태입니다. 여기에서 clock pulse와 Q의 숫자가 동일하기 때문에, 지금까지 binary 형태로 decoding을 거쳐야 했던 counter들과 달리 counting을 10진수 형태로 바로 알아낼 수 있다는 장점이 존재합니다.

(page 8)

세 번째로, UP down counter은, 지금까지 up 방향으로만 +1 count한 회로들과 다르게, down 방향, -1 으로도 count를 수행할 수 있는 회로를 의미합니다. up count 를 할지, down count를 할지는 Up/down 제어선을 통해 조절할 수 있는데, 제어선이 = HIGH 이라면 up count, = LOW 이라면 down count를 수행하게 됩니다.

(page 9)

이 표는 3-bit 에서 up count / down count의 순서를 설명하는 것인데, up count인 경우에는 0 0 0, 0 0 1 처럼 아래로 내려가는 형태이며, down count일 때는 0 0 0, 1 1 1, 1 1 0 형태로 올라가는 형태로 count를 진행함을 파악할 수 있습니다.

실제로 count = Q2 Q1 Q0 up / down counter은 JK flip-flop의 J = 1, K = 1인 toggle을 활용해서 구현하게 되는데, 과정은 아래 형태와 같습니다. (그냥 그대로 읽기)

(page 10)

사실 이 과정은 up인 경우 carry, down인 경우 borrow가 발생하는 조건과 동일하다고 생각할 수 있습니다. 아래 경우들은 toggle이 일어난 경우를 나열해 놓은 것인데, Q0는 항상 toggle이 발생하며, Q1은 Q0 + 1의 carry가 1일 때, Q0 – 1의 borrow가 1일 때 toggle이 발생합니다. 동일하게 Q2도 Q1Q0 + 1, Q1Q0 – 1에서의 carry, borrow에 따라서 toggle이 발생함을 파악할 수 있습니다.

(page 11)

따라서 이를 회로로 표현하면 다음과 같은 형태가 됩니다. 위쪽 부분은 up count일 때 Q가 사용됨을, 아래쪽 부분은 down count일 때 Q\_bar 가 사용됨을 파악할 수 있습니다.

(page 12)

네 번째로, Ripple counter은 비동기식 counter로 모든 flip-flop이 동일한 시간에 작동하지 않으며, 이로 인해 전파 지연이 많이 일어나는 counter 입니다. 따라서 전파 지연을 줄이기 위해서

(page 13)

다음과 같은 counter을 구현할 수 있는데, 이렇게 구현할 경우 전파 지연을 줄일 수 있지만, 회로가 더 복잡해짐을 파악할 수 있습니다.

(page 14)

다음은 추가적인 내용을 설명하겠습니다. 앞에서 설명한 shift register은 직렬 형태의 serial input, output 을 사용했었는데, 실제로 shift register은 input 혹은 output이 병렬이 되도록 설계할 수 있습니다. 앞에 보이는 회로는 output을 병렬적으로 만든 회로인데, 이를 통해서 Q0, Q1, Q2, Q3를 직접적으로 파악할 수 있으며, 직렬 형태의 data input을 병렬로 바꿀 수 있습니다.

(page 15)

두 번째로는 input이 병렬로 들어가는 경우입니다. 이 경우에는 앞에서 본 serial input shift register에서 모든 data가 저장되기 위해서 4번의 clock을 거쳐야 했지만, 이 경우에는 load 를 통해서 바로 data를 저장할 수 있으며, 병렬적인 data를 직렬 형태의 data로 바꿀 수 있다는 점도 존재합니다.

(page 16)

마지막으로 bidirectional shift register 은 이전에 설명한 Up down counter와 비슷하게 shift를 왼쪽 / 오른쪽 방향으로 수행할 수 있도록 구현한 회로입니다. 앞의 counter와 비슷하게 Right/left라는 제어선을 통해서 shift 방향을 조절하게 되는데, 이 경우에서 특이한 점은, 이전까지의 회로들과 다르게 왼쪽으로의 shift를 구현하기 위해서 오른쪽 flip-flop의 output이 다시 왼쪽 flip-flop의 input으로 들어갈 수 있다는 점입니다. 회로에서 보면 Q1이 실제로 G5와 연결되어 가장 왼쪽의 flip-flop의 input으로 들어갈 수도 있다는 것을 파악할 수 있습니다.